The Dissertation Committee for David DeAngelis
certifies that this is the approved version of the following dissertation:

Encouraging Expert Participation in Online Communities

Committee:

K. Suzanne Barber, Supervisor

Aristotle Arapostathis

Luis Francisco-Revilla

Christine Julien

Dewayne Perry

Encouraging Expert Participation in Online Communities

by

David DeAngelis, B.S.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2011

To family and friends near and far, may this work bring us a tiny bit closer.

Acknowledgments

I would like to express my sincere appreciation for those who have helped me along the way. "Thank you" to my family and friends for their support, encouragement, and entertaining distractions over the past seven years. In particular, my fiancé, Melissa Lott, has been unfailingly supportive and understanding while I concentrate on my work.

My advisor, Dr. K. Suzanne Barber, has encouraged my research and provided a laboratory with the freedom to pursue my interests. I would also like to thank the members of my dissertation committee, who not only offered their time and expertise, but truly guided and taught me throughout this research.

I am thankful for the opportunity to work with such great people. Suratna Budalakoti has been an outstanding collaborator and sounding board, and Rick Scott has been an indispensable source of advice and entertainment. Seth Holloway, Robert Grant, Chris Jones, Karen Fullam, Jaesuk Ahn, Jisun Park, Nishit Gujral, Dave Han, and Dung Lam have provided valuable research discussion and guidance, and made my time in the lab much more enjoyable. I would also like to thank Melanie Gulick, Charlotte Harris, and Stephanie Cardenas for helping to make things happen. Thank you all.

Encouraging Expert Participation in Online Communities

Publication No. ___________________

David DeAngelis, Ph.D.
The University of Texas at Austin, 2011

Supervisor: K. Suzanne Barber

In concept, online communities allow people to access the wide range of knowledge and abilities of a heterogeneous group of users. In reality, current implementations of various online communities suffer from a lack of participation by the most qualified users. The participation of qualified users, or experts, is crucial to the social welfare and widespread adoption of such systems. This research proposes techniques for identifying the most valuable contributors to several classes of online communities, including question and answer (QA) forums and other content-oriented social networks. Once these target users are identified, content recommendation and novel quantitative incentives can be used to encourage their participation. This research represents an in-depth investigation into QA systems, while the major findings are widely applicable to online communities in general. An algorithm for recommending content in a QA forum is introduced which can route questions to the most appropriate responders. This increases the efficiency of the system and reduces

the time investment of an expert responder by eliminating the need to search for potential questions to answer. This recommender is analyzed using real data captured from Yahoo! Answers. Additionally, an incentive mechanism for QA systems based on a novel class of incentives is developed. This mechanism relies on systemic rewards, or rewards that have tangible value within the framework of the online community. This research shows that human users have a strong preference for reciprocal systemic rewards over traditional rewards, and a simulation of a QA system based on an incentive that utilizes these reciprocal rewards outperforms a leading incentive mechanism according to expert participation. An architecture is developed for a QA system built upon content recommendation and this novel incentive mechanism. This research shows that it is possible to identify the most valuable contributors to an online community and motivate their participation through a novel incentive mechanism based on meaningful rewards.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Untapped capabilities permeate large-scale networks. Search engines specialize in identifying existing static documents on a network that are appropriate for a given query. Online communities such as question and answer (QA) forums, discussion forums, social networks, and news aggregators provide a method of connecting users and resources that can leverage both the static and dynamic (live) capabilities of a network of human users.

This research focuses on identifying content creators, or experts with desired knowledge, and motivating them to contribute to the social benefit of the online community.

No single user has complete knowledge across many different domains. On a large enough network, however, it is likely that somebody has expertise in nearly every domain. This research proposes techniques to facilitate the voluntary flow of information from one human user to another. These techniques are combined in an architectural framework for a QA system that will allow users to answer questions best addressed by human experts, such as personalized recommendation. For such a system to be successful, it is essential that it be able to identify and access experts in any given area, connect the responder(s) to the original questioner, and motivate potential responders to participate. This research proposes novel techniques for identifying expertise, a recommendation-based architecture for matching content creators and consumers, and a quantitative incentive mechanism for encouraging expert participation in an online community populated by a set of greedy users, or agents.

Online communities are enabled by the prevalence of popular websites built upon social technology. Websites built on user generated content (UGC) are prevalent, and the perceived value of this content is growing rapidly [78]. Sites like Twitter, Yelp, Digg, Reddit, eBay, Yahoo! Answers, Amazon, and many others rely on content that is created by their users, whether it is product reviews and descriptions, restaurant suggestions, movie recommendations, or any other kind of information. Often these websites allow each user to create an online identity. Through contributions to the site, users build a reputation through the collective whole of other users. This reputation and its associated measure of trust form the essence of an online community.

An effective QA system must facilitate the sharing of information. A question is provided by a user called a questioner and any other user (called a responder) is capable of reading this question and providing a response. QA systems provide a forum for typically human users to post questions and others can provide responses. Some early QA systems have been in use since the 1950s and simply take the form of a regular column in a printed trade journal [26]. The most sophisticated QA systems today use data analysis techniques to pair questioners and responders [1]. This research leverages models of user expertise based on previous participation to drive a recommendation architecture that pairs questioners and responders. In addition, the experimental system uses a novel incentive mechanism based on extrinsic reciprocal rewards to encourage participation from the most qualified users. Questions are usually classified as open or closed. Open questions are questions that accept additional responses. Open questions become closed after a period of time or by the request of the questioner. The scope can be as narrow as a single highly specialized field, such as zero gravity combustion, or broad enough to include countless topics from mathematics, to music, to the culinary arts. Many QA systems, such as Yahoo! Answers [4], manage a broad scope by forcing questioners to label their questions into a certain category, such as philosophy. Each category essentially functions as a separate forum, with its own audience and participants. The only thing these category participants have in common is an interest, but not necessarily expertise, in the category.

Modeling the abilities of users in a network-based QA system requires a measure of certainty in the information content passed along such a network, as well as the capabilities and intent of the source of the information, the responder. Trust is a measure of the fidelity or truthfulness of a responder and can be established in two ways. First, repeated interactions coupled with direct observations can be used by the questioner to construct a trust model [41]. A responder that provides poor information or is deceitful has a low measure of trustworthiness. The second method for establishing a trust model is the exchange of reputation information [60]. One trusted agent on a network may have experience with another particular agent, while others may not. Such an agent can share the trustworthiness findings of its own experience, allowing participants to quickly bootstrap their own trustworthiness models of the responder. Trust is not limited to a single measure; multi-dimensional trust can be established based on the quality of the answer provided, timeliness of the answer, percentage of questions the responder is available to answer and more [31].

QA systems often do not explicitly address the concept of trust; it is assumed that participants are honest and have nothing to gain through deceit. Deceitful or poorly communicated information generates the same effect as an honest, though unsatisfactory response. Properly evaluating a response also requires a measure of expertise. A deceitful response may appear genuine to the original questioner, but another expert could identify malicious intent. Through a feedback driven incentive mechanism, a QA system can distinguish between fraudulent behavior such as spam and a simple lack of expertise and can adjust incentives accordingly.

Closely related to trust is the concept of an expertise metric. Instead of modeling the fidelity of the answer or the trustworthiness of the responder, expertise is a topic-specific measure of the abilities and will of a networked agent, independent of intent or trustworthiness. Topic specificity means an agent may contain high expertise in one topic, such as gardening, and low expertise in another, such as language skills. In the context of QA systems, expertise is defined as “the ability of a user to answer a given question to the satisfaction of the questioner.” [15] Like trust, this research develops expertise models in the following two distinct ways. First, the historical content supplied by an agent can be analyzed. Using standard data mining techniques, previous agent activity can be parsed and clustered into topics. An agent demonstrating higher activity levels among certain topics suggests expertise in that topic. Much like trust model evolution, a feedback loop indicating satisfactory or inadequate activity can be used to alter the expertise model for a specific topic. The second source of information for building expertise models is the connectivity graph between networked agents. If agent A has high expertise in topic X, agent B has an interaction with A concerning X, and A indicates a satisfactory experience with B, it can be concluded that agent B has expertise in topic X. This means that expertise models can be developed using the actual content of a networked agent’s activity as well as using the network graph structure in a feedback loop.

Beyond expertise and trust, is influence, a third analysis tool for QA which enables user comparisons in a complex networked environment. Influence is a function of expertise and the activity, or participation level, of a networked entity. A networked agent’s activity level can be ascertained by the number and weight of edges in a graph. The most influential agents are not always those containing the most expertise, nor are they always the ones with the most activity. Influential agents are those that have the most impact on network activity. In the context of a QA system, such activity includes creating highly rated answers to difficult questions, posting of insightful questions, or even quickly posting answers to simple questions.

Many QA systems are currently open for public participation. The largest such system, Yahoo! Answers (YA), boasts over 120 million users and more than 400 million answers [51]. These figures indicate a participation of only 3.3 responses per user. In reality, many users will post one or more questions and never supply a response to another. Moreover, the knowledge shared in Yahoo! Answers is very broad but not very deep [2]. Categories exist for topics as diverse as astronomy, celebrities, relationships, computer programming, and many more. In a study performed by Adamic et al., only 1% of the questions in the Programming category required expertise above the level of a student with a single year of experience [2]. This indicates a lack of question depth in YA. It is possible that this shallow depth drives away users with higher expertise because they feel they have nothing to gain by responding to shallow questions, and they lack confidence in the abilities of the responders.

Smaller, more specialized QA systems have been developed to combat this shallow question phenomenon and draw out valuable experts. One such example, Experts-Exchange (EE), focuses on computer and information technology related questions. Its users are mostly young to middle aged males in the IT sector [67], and it contains over 17 million solutions from 260,000 contributors and 3.9 million viewers [37]. EE makes a distinction between members and contributors, or experts. Members must pay a fee to access the archived solutions and ask new questions. Experts must periodically participate by answering questions to maintain their status as an expert as well as their archive and ask privileges. An expert's solution is evaluated by the questioner, which influences an expert's status. The mechanism driving EE is based on the idea that people are willing to either pay money (and become a member) or satisfactorily answer questions (and become an expert) in exchange for the ability to ask questions and access the archive. This type of reward is called a *systemic reward*, meaning the user is provided with something of value within the context of the system in exchange for valuable participation. Choosing a reward with meaningful value is an important part of the process of designing an incentive mechanism, and this is discussed more thoroughly in section 5.1.

A key characteristic of the Experts-Exchange mechanism is that a credit value is maintained. This credit can be earned and spent by responding to and asking questions [21]. A rapidly growing QA site named Stack Overflow yields the quickest responses of any major public QA site [55], and the designers of Stack Overflow extend this idea in the form of bounties. A bounty is an additional number of reward points that the question asker offers in exchange for a satisfactory answer. This sort of mechanism may allow a user to garner a large set of responders for an urgent question, but it is not without drawbacks. This spending of credit discourages a user from asking questions. This questioner inhibition causes many questions to never be asked, even though they may be of great value to other users once answered. The incentives for online communities developed in this research are based on meaningful rewards, but influence cannot be *spent*. Also, the proposed system combats the shallow question phenomenon not by restricting scope, but by recognizing and encouraging expert participation across all disciplines.

1.1 Problem Statement

Many different online communities, including question and answer systems are currently available, but they all suffer from a similar set of problems. First is a lack of participation. It is beneficial to encourage expert participation from users in order to reliably secure valuable content, thus directly adding value to the community. In addition to content creator participation, consumer participation is also desirable. In the context of QA systems, formulating a detailed question requires skill and effort, and the presence of richer and more varied questions draws out the expertise of the responders. Spending credit when asking a question discourages frivolous questions, but it also does so at the expense of less content on the QA system. A major barrier toward participation is the time investment needed by a content creator to find an appropriate piece of content to create or an appropriate question to answer. The most highly qualified experts are likely to be very busy, therefore they are often unwilling to browse or search through a large collection of open questions until they find a suitable match. This research addresses this lack of participation by providing meaningful incentives for participation and by making it easier to participate in a constructive manner through a recommendation-based architecture.

A second problem with current online communities is a lack of confidence in the expertise and trustworthiness of the content creators. For example, if a question is viewed by only the most recent visitors or those who chose to scan a long list of questions, the questioner has no reassurance that those who respond are qualified or honest. Current systems reward certain behaviors, such as answering a question, and this reward is in the form of a user score. This score does not accurately represent the value of a user in the QA system because the score is created by simple actions or evaluations by the original questioner, who may not be best qualified to judge the value of a response. This research addresses this lack of confidence through a recommendation-based architecture that matches questions with qualified experts and a collaborative aggregative incentive mechanism that uses the collective peer expertise to reward the most valuable contributors. Because a question is routed to those who have demonstrated experience on that topic, a questioner can be assured that his question has been seen by some of the most capable potential responders.

A third problem with current online communities is that they suffer from various social phenomenon such as nepotism, reciprocity, and bandwagon effects. This is discussed in section 2.2.3. The social mechanisms behind online communities are intended to drive higher participation, increase user retention, and build a sense of community among the user base. Often these social mechanisms backfire and have unintended consequences. Section 4.2

describes an algorithm for unbiased trust estimation. This measure of trust is designed to counter the deleterious effects of social mechanisms in online communities, and allow the most valuable content creators to be identified.

1.2 Research Questions

This research will examine the following hypothesis:

In an online community, a non-monetary quantitative incentive mechanism applied to a recommendation-based architecture can increase expert participation and satisfaction - both as a content creator and a consumer - and assure confidence in the value of the provided answers.

The hypothesis asserts that a quantitative incentive mechanism, built from observing the behavior of users, can effectively increase beneficial system-wide properties such as expert participation as well as accurate, timely, and trustworthy content generation. The mechanism is based on the idea of reciprocal systemic rewards. These rewards are given to those demonstrating the highest levels of expertise, and they in-turn have the greatest systemic influence over others. Thus, the most influential users are rewarded by others' prompt, satisfactory responses when they pose a question. The reward is designed to encourage expert participation. The hypothesis is tested by the following research questions.

1.2.1 Research Question 1 - Measuring Expertise

RQ 1: How can the expertise of a user in an online community be measured?

In an online community made of human volunteers, some users will create more valuable content than others. Some users may have proficiency in a certain topic - such as books and authors - while others may demonstrate proficiency in another domain. A fundamental tenet of this research is that nobody knows everything, but everybody knows something. The first step toward encouraging expert participation is identifying those who are considered experts. Recommendation is closely related to expertise identification because some users may be desired for their expertise in certain domains but not in others. An effective recommendation engine can match content creators with content consumers in order to fulfill a need, regardless of some general expertise metric. Such a recommendation engine is described in section 4.1

Expertise can be identified using techniques described in chapter 4. Expert responders are highly desirable in a QA system because they are capable of answering very difficult questions that others cannot. Expertise is identified through a multi-dimensional topic-specific expertise model [15]. Essentially these techniques draw from two sources of information: content-based information and link-based information. Content-based expertise identification techniques involve analyzing the text content generated by users in an attempt to ascertain the skill level of the user. Link-based expertise identification techniques use the underlying graph structure generated from the social interactions of the users within the community. It is important to note that simply measuring node properties such as *in degree* in a question and answer graph are insufficient for determining the true expertise level of a user because social effects often outweigh the value of the content produced by a user. Section 4.2 describes a more sophisticated method for identifying experts based on the link structure of a social network.

1.2.2 Research Question 2 - Choosing an Incentive

> *RQ 2: How effective are systemic reciprocal rewards for encouraging expert participation in an online community?*

$RQ1$ discusses how to identify experts in a community. Once this target demographic is identified, we must encourage their participation for the community to maximize its value to its own users. One way to encourage participation is to lower the barriers of entry. The recommendation system presented in section 4.1 frees a user from searching through a database of potential questions in order to find one that is appropriate to answer.

Beyond this, various incentive mechanisms have been used in online communities to encourage participation. Fundamentally, an incentive mechanism rewards a user who exhibits socially beneficial behavior by giving him/her something of value. Yahoo! Answers and many similar sites reward contributors with arbitrary *points* for positive contributions. Additionally, they may offer a leader board position, or virtual trophies and badges. For many users of such systems these rewards are meaningless. This research attempts to identify a reward that is valuable to a larger portion of the users. Such meaningful rewards are called *systemic rewards* because they add value within the framework of the system. They are designed to give extra functionality, enjoyment, or ease of use to the awardee. For example, in a QA system a potential systemic reward is priority access to potential responders when asking a question. This reward is said to be *reciprocal* because the reward gained by a content creator is a function of the social influence of the user who requested the content. In the context of QA, a question asked by a user in high standing is "worth more" than a question asked by a user in lower standing. Section 5.1 presents a survey that was administered to evaluate the desirability of such a reward for QA systems. This research direction attempts to demonstrate that access to peer generated content can directly motivate people to apply their own expertise, thereby generating more content.

1.2.3 Research Question 3 - Mechanism Design

> *RQ 3: In the context of a question and answer system, how can an incentive mechanism be designed to encourage expert participation?*

*RQ*1 raises the question of identifying desirable experts in an online community. *RQ*2 lays the foundation for encouraging their participation by establishing a meaningful reward. *RQ*3 concerns fully developing and measuring an incentive mechanism based on such a reward. Once a reward is chosen, it must be carefully distributed to the users in order to encourage positive behavior and discourage negative behavior. Such a mechanism is said to be *incentive compatible*.

Some examples of beneficial responder behavior in QA include the creation of prompt, relevant, and correct (where applicable) responses. Examples of beneficial questioner behavior include asking rich questions, such as asking for advice pertaining to a detailed situation. One of the greatest strengths of QA is that humans are capable of answering more sophisticated questions than those most suited to an internet search query. The incentive mechanism must be designed to encourage such beneficial behaviors.

In addition to encouraging beneficial behavior, the incentive mechanism must discourage harmful behaviors. First, false or biased responses to questions can cause harm and should be minimized. Second, spam is a major concern for internet-based QA systems. While perhaps not as harmful as false information, unintelligible responses are of no use. This could be caused by something as straightforward as language issues, or unintelligibility can be symptomatic of something deeper, such as a large disparity of expertise between the questioner and the responder for the question topic. A third point of concern for many QA systems is each user's question to answer ratio. A forum full of questions with very few answers is of little use. Likewise, a QA system with too few questions is underutilizing the skills of its user base. This research does not propose discouraging questions; it proposes encouraging strong responses.

A QA system incorporating a recommender coupled with the proposed incentive mechanism could facilitate such behavior by recommending appropriate questions for a responder in order to minimize the time investment spent searching for a question to answer. This would encourage more responder participation by experts. Another beneficial effect of such a QA system is that questioners would have increased assurance in the accuracy and honesty of the responses. Incentives in currently available QA systems do not encourage honest participation (both asking and responding to questions) by experts. They assign points for very simple actions. The research in this dissertation allows the community to decide which content is most useful, with experts weighing in more heavily.

Section 5.2 presents an incentive mechanism for QA systems and demonstrates its performance in a software agent-based simulation. This simulation was created to model the behaviors of real users in several different QA systems today, including Yahoo! Answers and the Java Forum. This simulation directly compares the performance of a leading industry incentive mechanism to the incentive mechanism presented in this research. The proposed incentive mechanism is designed to preserve beneficial system-wide behavior while each agent attempts to maximize its own utility, or influence. It has been established that humans do not always behave in a rational manner [43], whereas software agents do so by definition. However, section 5.1 establishes that systemic reciprocal rewards are in fact desirable to the majority of users, and it is expected that they will behave rationally.

1.3 Contributions

This dissertation will form the foundation for a new class of online communities. Such systems will not rely on monetary payment for participation but will reward contributors with influence in the system. Those who contribute the most value will be rewarded with the effort of others, in the form of priority given to their own questions. Nobody knows everything, but everybody knows something. A professor of computer science may answer a very difficult question on complexity theory, but then have a simple question about gardening. This user has valuable expertise, and he or she is a desired participant. He or she will be rewarded when asking a simple question about gardening because a responder will earn greater rewards when answering his or her question than another. Not all questions are difficult. Many do not require very specialized or broad expertise. Therefore, it is desirable to reward participation, even if the participant is not an expert. For example, a teenager may not have much expertise, but he may have plenty of free time to spend answering questions. The incentive mechanism in this research rewards such behavior.

In developing this QA system, several basic contributions must be accomplished. First is a multi-dimensional topic-specific model of expertise. This model of expertise is useful for identifying desirable contributors to a QA system. Section 4.2 introduces another model of expertise and trust that is applicable to social networks in general. These models are a necessary step toward calculating the influence and desirability a user has in the system. The second contribution of this research is the concept of systemic rewards. Choosing a reward that is meaningful to the largest group of users is of key import when building a system designed to encourage the strongest expert participation. The third contribution is the full implementation of an influence-based incentive mechanism for encouraging beneficial behavior. The fourth contribution is a novel architecture for a recommender based question and answer system using the proposed incentive mechanism. More generally, this research is the first to show that peer generated content can be effectively used to encourage expert participation in online communities.

The power of online communities is unrealized. Question and answer systems are a great example of underutilized online communities. Internet search has become the norm, but many questions are unsearchable using today's paradigm of feeding keyword-based search queries into a search engine and then digesting the results. Such questions are often concerning advice, recommendations, or very specialized knowledge. Even a question as innocuous as choosing which bicycle to purchase can be very difficult. Using a search engine, a person would have to decide which type of bike suits his needs by reading tutorials on bicycle purchases, and then he would have to search the inventory of several different manufacturers and match this list with the pricing and availability at local bicycle shops. Once a short list is created, he would have to compare each bicycle before making a decision. With an effective QA system, an expert can give a model name that would be well-suited to the questioner's requirements, perhaps with localized content such as price, availability, or alternative models. In the first scenario only the original questioner had to expend effort while in the second scenario both the questioner and the responder(s) did. With an effective QA system, the total sum effort expended is significantly less. This research facilitates this type of interaction by encouraging expert participation through new techniques for identifying experts and a novel non-monetary incentive mechanism.

1.4 Outline

The purpose of this research is to encourage expert participation in online communities. This document discusses the full stack of tasks needed to accomplish this goal, including identifying a target set of experts, choosing an incentive, and distributing rewards in an efficient manner. Experimental support is provided within each section introducing new technology.

Chapter 2 contains a discussion of related work and how the research proposed here differs and complements existing work. The chapter opens with a discussion on motivation and the theory behind the incentive mechanism. Next, several different QA systems are discussed, along with their respective strengths and weaknesses. Specifically, this chapter includes discussion of the currently available incentive mechanisms that drive existing QA systems. The QA system proposed by this research attempts to address their shortcomings while preserving their benefits. In addition, chapter 2 discusses the current state of the art in expertise modeling and recommendation. The roles of trust in forming expertise models and in social networks are also discussed.

Chapter 3 introduces a high level architecture for a novel question and answer system. A question and answer system is a prototypical online community, and this serves as a concrete application domain for the technology developed in this dissertation. This architecture is referred to throughout the dissertation and serves to illustrate how the different pieces of technology interact to form a cohesive system. This is followed by a discussion of the architecture from the viewpoint of a user and again from the viewpoint of a recommendation agent.

Chapter 4 describes techniques for identifying experts in online communities. First is a technique suited for QA sites based on a generative model of user behavior. This technique utilizes both link-based information and content-based information. Following this is a technique for identifying the most trustworthy expert users in another type of online community, a news aggregator. Both of these techniques are presented with experimental support.

Chapter 5 discusses the design and development of a novel incentive mechanism for online communities. Early in this chapter, section 5.1 evaluates the results of a survey designed to investigate the choice of reciprocal systemic rewards. The remainder of the chapter discusses the design of a full mechanism and the performance of a software simulation of a question and answer system operating on this mechanism.

Chapter 6 states the contributions of this research and addresses the original hypothesis and research questions. This chapter also describes the impact that this work may have on the design of future online communities.

Chapter 2

Related Work

Effectively encouraging expert participation in an online community has several steps. First, human behavior must be understood as it pertains to peer production systems such as web-based QA [8]. Why do people participate in question and answer systems (QA)? Is there some perceived reward that will be received in exchange for participation? Does the act of participating in a QA system become its own *intrinsic* reward? Second, we must identify our target audience. Who do we want to encourage participation from? What kind of behavior are we looking for? Are there any behaviors we wish to discourage? Social roles are useful for answering these questions and identifying a target participant. Social roles are described as characteristic communication patterns between network members, or a "structural signature" [84]. Once an idealized target audience is identified, it is necessary to model the population of system users and selectively reward those who exhibit beneficial behaviors. These user models can take many forms, and this work proposes a topic-specific expertise model, which is rooted in the trust in multi-agent systems research community.

When paired with participation models, these expertise models can communicate the value of a user in the community. It is imperative to identify the most valuable users and encourage their participation in order to maximize the social welfare of the system. Many existing QA systems attempt to identify and reward positive behaviors in users, but they typically rely on questionable rewards and very simple user models. This chapter will establish the foundation of the proposed incentive mechanism based on collaborative aggregation. With the expertise models and incentive mechanism in place, it is possible to construct a new type of QA system based on recommendation. Questions can be routed to highly appropriate users with such a system.

This chapter will discuss classic and current research in human motivation, along with properties of desirable system contributors. This is followed by a section on related work concerning user modeling. This chapter concludes with sections on current recommendation and incentive mechanisms, specifically as they apply to QA systems. Examples are drawn from the current state-of-the-art in question and answer systems.

2.1 Motivation

An incentive mechanism requires an incentive. That is, the core of an incentive mechanism is something of perceived desire that is used to shape behavior, or motivate. In psychology research, motivation is often divided into two distinct categories: intrinsic and extrinsic. Intrinsic motivation is caused by the pleasure of performing an activity [86], whereas extrinsic motivation requires an expectation of something desirable from outside the performer.

Psychologist Steven Reiss has stated that extrinsic motivation may be classified as *drives* such as hunger, while intrinsic motivation is typically associated with intellect, such as curiosity, autonomy, and play [68].

2.1.1 Extrinsic Motivation

Before we can understand what will motivate a person, or what a person wants, we must understand his needs. In his 1943 seminal work, Abraham Maslow defined a hierarchy of human needs [56]. Much research in motivation is based on the satisfaction of these needs. The base needs are physiological and safety needs, while the higher levels of belonging, esteem, and self-actualization are most pertinent to QA systems. It is possible that people participate for a sense of belonging to a community, but it is also likely that they are fulfilling a need for esteem. This would include a need for confidence, achievement, and respect by others. Sometimes this sense of fulfillment is enough to motivate the users of a QA system to devote their entire time on a QA system toward answering the questions of others while asking no questions of their own [83]. These esteem needs serve as an extrinsic motivator because they generally rely on acceptance from others.

A more immediate type of extrinsic motivator would be to directly and explicitly reward beneficial behavior. Prior to the proliferation of online communities, much work has been done in the study of mail-in survey responses. Public opinion and medical researchers have discovered numerous properties that affect survey responses, and some are completely obvious such as survey length or monetary incentive, while others are more obscure, including commemorative versus ordinary stamps and colored ink [24] [6]. These studies indicate a positive correlation between survey respondents and specific extrinsic rewards.

A simple effective extrinsic reward is a monetary incentive. Mizes, Fleece, and Roos have shown that monetary incentives increase response rate while biasing response rates little [58]. More surprisingly, they have also shown that larger incentives do not necessarily increase the response rate [58]. It has also been shown that if a monetary reward is offered, it must be made clear to the responder during the first time of contact for greatest impact [11].

Monetary incentives have been applied to online QA systems. The now defunct Google Answers was the first, while Uclue, Mahalo Answers, TaskCN, and others continue today [14] [38] [79] [87]. These services sometimes charge a small upfront fee for asking a question, then the questioner can offer up to $400 for an acceptable answer. Typically the service will claim a percentage of this answer fee. While such a model can sustain a business given the right audience, the fees also create a barrier to participation. Yahoo! Answers contains over 1 billion questions and answers [77], and its free service model has certainly contributed to its widespread adoption. Yang et al. have observed on taskCN that of 1.7 million users, only 3.2% have ever won [87]. Additionally, expertise has little to do with who wins the rewards; winning is based almost entirely on competition level. Those who win rewards adopt a strategy of specifically targeting only the most unpopular questions, regardless of content [87]. Moreover, recent research by Chen et al. shows that higher prices for answers leads to significantly longer but not better answers [17]. Additionally they stress the importance of reputation systems through their observations that a responder with a higher reputation provides significantly better answers. This concept of modeling users is integral to the QA system proposed here and will be further explained in chapter 4.

Non-monetary extrinsic rewards are the most common type of motivator in QA systems today. Such rewards have also been investigated in the context of mail-in survey responses. Church has observed that non-monetary rewards yield a comparable response to monetary rewards, and the timing of the reward (upfront or conditional on completion) has a much greater effect than the reward type [18]. Nederhof has investigated non-monetary rewards for survey participation and found that a simple ball-point pen dramatically increases user response when the response rate is small [62]. Subsequent mailings without an incentive converge toward the control rate. Interestingly, the effects of giving a token non-monetary incentive dwindle as the response rate increases. This is highly applicable to the standard bulletin board model for QA systems. In theory, each of the millions of users can respond to all of the millions of questions. In practice each user provides a total of 3.3 answers [51], meaning Yahoo! Answers has an extraordinarily low response rate, despite confounding variables such as practicality and readership. This would indicate that non-monetary extrinsic rewards are appropriate for encouraging participation in QA.

Table 2.1: Yahoo! Answers reward structure

Action	**Points**
Begin participating on Yahoo! Answers	One time: 100
Ask a question	-5
Choose a best answer for your question	3
No best answer was selected by voters	Points Returned: 5
Answer a question	2
Deleting an answer	-2
Log in to Yahoo! Answers	Once daily: 1
Vote for a best answer	1
Vote for No best answer	0
Your answer selected as best	10
Receive a thumbs up on a best answer	1 per, max 50
Question removed due to violation	-10

The most popular QA site, Yahoo! Answers (YA), relies on such a system of extrinsic non-monetary rewards [77]. As the market leader, many other sites have emulated the point based incentive system found on YA. Some fundamental criticisms of these incentive systems are that points have no real world value, they cannot be traded, and they simply indicate the activity level of a participant. The only value that such points may have is esteem based, or simply put, bragging rights. A fundamental benefit of the work in this research is that *points matter*. Section 2.1.3 introduces the motivation behind the proposed incentive mechanism, which is further described in chapter 5.

Many different techniques exist for quantifying the value of a contributor. Table 2.1 describes the reward structure currently used in Yahoo! Answers [39]. Jain, Chen, and Parkes have proven that the best answer voting rule in YA is theoretically sound; that is, it effectively rewards beneficial behavior [40]. However, they have identified two different scoring rules that improve the efficiency of the game equilibrium. Even if the rules are theoretically correct and optimal, they are still used to distribute points that have no *value*. Experts-Exchange (EE) is a rare exception to this rule. EE awards points for answering questions, and points can be redeemed for membership [37], which can also be purchased. Membership allows a user to ask questions to others, so this privilege is a form of non-monetary extrinsic reward.

2.1.2 Intrinsic Motivation

In contrast to extrinsic motivation, intrinsic motivation is driven by needs or desires that come from within. The highest level of Maslow's need hierarchy, self-actualization, can become a strong motivator for QA users. This level includes the need for creativity and problem solving [56]. These needs on the self-actualization level often serve as intrinsic motivators because they originate from within, whereas the lower levels serve as extrinsic motivators because they rely on acceptance from others or other forms of reciprocity.

There is strong evidence against the existence of a strict needs hierarchy [82], but the classification of needs remains useful and relevant to modern motivation research. Reiss has developed a listing of sixteen basic desires, and the intrinsic, intellectual desires are most interesting in the context of QA systems [68]. Some of these basic desires include curiosity, idealism, independence, and power. It is easy to imagine a situation where one or more of these desires may drive a person to participate in an online community.

The fundamental identifier of intrinsic motivation is that the task itself is rewarding. A simple example of intrinsic motivation is a person playing a game. He usually expects no reward other than the enjoyment that is provided by participating in the game. While this may seem unproductive, games have been created in which the player contributes a useful service to others. One class of games, called Games With a Purpose (GWAP), enables people, as a side effect of playing, to perform tasks that computers cannot [81]. Some examples of such tasks that people may find enjoyable include image tagging, audio tagging, and ontology creation.

Intrinsic motivation is very pertinent to designing a question and answer system. If participation yields its own rewards, participants will be more likely to return. These intrinsic rewards are manifested in a sense of achievement. In fact, it is likely that some of the most desirable experts are motivated intrinsically. Smith has discussed that many of the most successful responders in Yahoo! Answers spend an average of several hours per day answering questions, without ever asking a single one [83]. Because the tangible value of the extrinsic reward structure of Yahoo! Answers is minimal (see Table 2.1), it can be assumed that these most prolific experts are motivated intrinsically.

2.1.3 Motivating Q&A

Both intrinsic and extrinsic motivators can be enormously useful in eliciting participation in a QA system. People respond differently to different stimuli, so it behooves a system designer to address the broadest possible population. The most valuable experts, those who spend hours daily giving high quality answers, are an elusive group [83]. Instead of focusing solely on this group, who most likely are motivated intrinsically, this research attempts to cast a wide net and encourage participation from those who respond to a wider variety of motivators. This research addresses both intrinsic and extrinsic motivation for participating in QA systems.

The agent-driven recommendation based QA architecture is designed to function as an intrinsic motivator. The goal of this architecture is to transform the process of answering questions into a fun, enjoyable challenge. The recommendation of potential questions to answer streamlines the mundane operations of searching for a suitable question and evaluating if it is worth the time needed to answer correctly. From a questioner perspective, confidence in a prompt, satisfactory answer can increase the enjoyment of participating in such a system. While the desirable properties of the answers themselves form an extrinsic motivator, the enjoyable process of participating in the system yields intrinsic rewards. The recommendation process described in section 4.1 is designed to appeal to intrinsic motivation. One current QA system, Aardvark, leverages intrinsic motivation by using a recommendation system for connecting questioners to responders [1]. In Aardvark, responders are matched to questioners by analyzing the question and searching through friends' profiles on a social network to find a match. Blurtit is a QA system that also utilizes an intrinsic motivator; simplicity. While the incentive structure is not fundamentally different from its competitors, Blurtit is designed to have minimal barriers to entry by not requiring registration and limiting questions to 20 words [12].

This research takes a more pragmatic stance on extrinsic motivation. At its core, the incentive mechanism relies on the following assumption:

> *A user will be motivated to answer the questions of others if he can be assured of more satisfactory answers to his own questions.*

This assumption is based on the concept that everyone knows something, and nobody knows everything. Even the most accomplished and knowledgeable people will have questions outside of their own area(s) of expertise. This idea of eliciting a satisfactory response in return for participation creates an extrinsic reward structure. Through the principle of reciprocity, a participant expects something desirable from another in exchange for providing satisfactory answers. This mechanism for evaluating and distributing extrinsic rewards, or *influence*, is further described in chapter 5.

Such an approach toward extrinsic motivation has been successful in peer production based projects such as software development. Peer production is a method of solving problems where individuals who have the best information available about their own fitness for a task can self-identify for the task [8]. One such project is the development of the Linux operating system. The project has been enormously successful, due in large part to a team of highly skilled and motivated volunteer developers. Kollock posits that their motivation for creating digital public goods stems from anticipated reciprocity [47]. He states that one may contribute to the group with the expectation that others will do the same, creating a generalized exchange and system of credit where one can draw upon the effort of others without the need to immediately reciprocate [47]. Not all peer production systems are entirely benevolent; Pouwelse et al. explain that many systems contain "pirates and Samaritans" [66]. Pirates may add value by illegally sharing content at the expense of the content creators. Nevertheless, Kollock shows that using a relaxed accounting strategy that allows for some slack in repaying debts has many advantages that outweigh the increased vulnerability to exploitation, including the ability to dampen cycles of recrimination, especially when interacting with a subset of known individuals [48]. This is relevant to the proposed QA system for several reasons. First, in the proposed QA system it is not necessary to build credit before asking a question. Anyone can ask even the most difficult question regardless of their past participation. Secondly, giving a poor answer does not appreciably hurt a responder's status. The system is designed this way to encourage maximum participation. Additionally, Kollock stresses the importance of identity persistence and reputation [47]. Persistence is necessary for reciprocity to take place. Without a stable body of users, there is temptation to take advantage of the system because there is little lasting consequence. The proposed system is designed to facilitate reciprocity while maintaining identity and reputation.

2.2 User Modeling

The beginning of this chapter has been devoted to identifying potential motivators and characterizing desired participants in a recommendation-based question and answer system. Once this idealized target is known, it is necessary to select individuals from the pool of active participants that match this ideal most closely and incentivize them. This involves modeling each participant in terms of their relevant traits, such as expertise breadth and depth, along with their QA participation habits. Moreover, a recommender-based QA system requires user expertise models in order to manage the recommendation of questions to users and potential responders to questioners. Such modeling techniques are fundamentally based on the concept of trust.

Many popular QA systems model users, but they rely simply on participation metrics. Yahoo! Answers models users as a single integer, which is the sum number of points they have earned [39]. Aardvark is an exception, and it creates more sophisticated models based on user information stored in social networking websites [1]. Like Aardvark, the QA system developed in this research models the capabilities of its users in an expertise model. Unlike Aardvark, it uses trust based modeling techniques to discover breadth and depth of expertise across all topics.

2.2.1 Trust

Trust is a term that has many different meanings in different communities. In sociology, trust is a belief in the good character of one party [28]. The party is believed to seek to fulfill policies, ethical codes, and previous promises. This definition of trust goes beyond the idea of confidence. Confidence is the belief that a person or thing (or agent) is capable. An agent may be capable, but still possess no benevolent intent. This concept of intent is a central tenet of agent theory, and for this reason the agent research community has developed its own concept of trust. The concept of agent-based trust allows agents to predict the behavior of their environment with known certainty. Often, as presented here, trust has a statistical foundation.

Let E be an event, and let F be the set of properties that define the event.

$$Event\ E$$

$$E \leftrightarrow [F = \{f_1, f_2, \ldots f_x\}]$$

An agent or team of agents can make a prediction of event E, called P. This prediction has properties $p_1 ... p_x$ that approximate the properties of the event E.

$$P = \{p_1, p_2, \ldots p_x\} \qquad\qquad P \approx F$$

$$\max(trust) = \min(\Delta(P, F)) \tag{2.1}$$

Trust is a collection of techniques that allow an agent or team of agents to minimize the error between P and F through observation of historic behavior and communication. When an agent has the ability to predict the behavior of its environment, it is more capable of achieving its goals.

2.2.2 Multi-Dimensional Trust

Basic trust is used to simply minimize the Δ between actual and predicted system behaviors. Multi-dimensional trust (MDT) is a richer variant of basic trust in which an agent considers its goal state, as defined by its reward function, and behaves in a manner to maximize its goal achievement and resulting reward [31]. MDT is a technique for building trust models according to multiple dimensions that are specified by a domain problem and consequently an agent's reward function. Gujral et al. have demonstrated that, in many circumstances, MDT can be used to guide decision making resulting in higher goal achievement [31]. In a sense, MDT is a more applied version of basic trust. The MDT algorithm was developed to solve the partner selection problem, where a rational agent is incapable of solving a problem in isolation; therefore it will seek help from a potential partner agent. These agents are guided by a utility function that attempts to maximize goal achievement. This is analogous to a situation faced by humans in online question and answer systems. A questioner is incapable (or unwilling) to solve a problem alone and therefore seeks assistance from a suitable other user. The MDT algorithm below forms the foundation for expertise modeling and recommendation as described in chapter 4.

There exists a rational agent, A, and a set of potential partner agents,

P, with which the agent can interact.

$A = rationalagent$	where A behaves according to a strategy
$P = \{p_1, p_2, ...p_j\}$	where P is the set of all potential partner agents, from p_1 to p_j
$B = \{b_1, b_2, ...b_j\}$	where B is the set of partner constraint functions
$\forall p_z \in P, \exists b_z \in B$	each potential partner agent has its own constraint function
$b_z = \{f_{z1}, f_{z2}, ...f_{zr}\}$	

Each constraint function, b, is a set of functions (constraints) that govern the behavior of a potential partner z. These functions are expressed in the same domain terms as the reward function. The partner constraints considered for a QA system might include solution *quality*, *timeliness*, and *availability*.

$$R(f_1, f_2, \ldots f_s) = \text{reward function}$$

The reward function, R, is specified by the domain. The reward is a function of the *goal requirements* f_r, which are analogous to the partner constraints b_z, though not all partner constraints may be considered in the reward. This reward function determines the amount of reward (measured in reward units) earned by the rational agent A. The single subscript indicates that the reward supplied is dependent only on the performance of the partner agent, not its identity. In terms of a QA system, the reward represents the value of an answer provided to a questioner. It is a function of several properties of the answer, but the identity of the responder does not affect the value of the response. Let

$\{g_{z1}, g_{z2}, ...g_{zr}\}$ represent the *dimension* components in the behavior model of agent z. The following implication states that the number of dimension components in a behavior model must match or exceed the number of goal requirements for developing a generally applicable model.

$$[R(f_1, f_2, ...f_s) \wedge (b_z = \{g_{z1}, g_{z2}, ...g_{zr}\})] \Rightarrow (r \geq s) \tag{2.2}$$

In regards to QA, 2.2 states that the dimensions in the behavior model, r, must include all relevant congruent properties, s, of a potential responder, such as correctness, promptness, conciseness, and others.

The rational agent A maintains a model, b_z^*, of the constraint function b_z for every potential partner agent $p_z \in P$.

$$B^* = \{b_1^*(n), b_2^*(n), ...b_j^*(n)\} \qquad |B^*| = j \qquad \text{where } n \in \mathbb{N}$$

$$\forall p_z \in P, \exists b_z^*(n) \in B^*$$

$$\forall [b_z = \{f_1, f_2, ...f_r\}] \exists [b_z^*(n) = \{g_1, g_2, ...g_r\}]$$

The rational agent A maintains a set of behavior models, B^*, for every potential partner agent in P. Each constraint function that defines a potential partner's behavior, f_r, is approximated by a *dimension* component, g_r, in the behavior model. Because these models evolve, the subscript n indicates the state of the model at time n. B^* contains models, $b_z^*(n)$, of an individual potential partner agent z at time n. The cardinality of B^* is the number of potential partner agents, j.

This states that a rational agent maintains a model of the behaviors of each potential partner agent. In a recommender-based QA system, expertise models can be maintained by individual agents that represent a human user or by a single centralized recommender (see chapter 3). With distributed expertise models, many different models of the same entity are in use simultaneously. A simple scheme for propagating these models to maximize the social benefit of shared experience under communication uncertainty is described in [22]. While such a system of distributed expertise models is robust toward system failures, the communication overhead for maintaining n^2 models in a QA system becomes impractical. Therefore, the proposed recommender-based QA system uses a centralized system to model expertise, as described in chapter 3.

$$b_z^*(n) = \{g_{z1}, g_{z2}, ...g_{zr}\} = \left\{ \frac{\sum_{k=1}^{n-1} f_{z1}}{int_z}, \frac{\sum_{k=1}^{n-1} f_{z2}}{int_z}, ... \frac{\sum_{k=1}^{n-1} f_{zr}}{int_z} \right\} \tag{2.3}$$

$b_z^*(n)$ is the model of potential partner agent z at time n, and g_{zr} is the approximation of the constraint factor f_{zr} for potential partner agent z. int_z is the number of observed interactions with agent z. This means that the model g_{zr} is simply the average of the observed values of f_{zr} over all previous interactions during time 1 to $n-1$. This simple model updating scheme is sufficient when trust is represented as a normalized collection of continuous values and all interactions are equally weighted. The proposed QA system uses a more sophisticated updating mechanism for expertise models based on user feedback as described in chapter 4.

In multi-dimensional trust, these models are used to maximize the reward earned by an agent when selecting partners. The net reward earned by the agent A when choosing a partner z, and z is available, is defined as:

$$\text{reward} = R(b_z) - c_f - c_{zp}$$

Where c_f is the fixed cost of interacting with a partner, and c_{zp}, is the partner cost constraint. c_{zp} can be considered a service charge levied by the partner z. Therefore, the net reward earned by A is a function of the partner constraints (behavior characteristics, b_z) less an overhead cost and a service charge.

Agent A does not have direct access to the partner constraints, B, necessary to make the optimal choice of partner agent, though A does know the reward function R. In order to choose the best potential partner agent, agent A must consult its partner behavior models, B^*, and construct an *estimated goal payoff* ($EGP_z(n)$) for each partner agent z at time n.

$$EGP_z(n) = R(b_z^*(n)) - (1 - a_{z,est})c_f - c_{zp,est} \tag{2.4}$$

Where $a_{z,est}$ is the estimated availability of the partner agent, and $c_{zp,est}$ is the estimated cost constraint. These estimates are calculated just like any partner constraint function model g as shown above. Essentially, 2.4 is an expected utility function that is used to calculate the predicted utility for choosing a particular partner z at time n.

$$\exists x < j | EGP_x(n) \geq EGP_{\bar{x}}(n)$$

There is some potential partner agent x at time n that yields the greatest estimated payoff for a given, known reward function. This agent is then selected and the rational agent A receives the reward shown above.

This method of choosing a partner based on estimated performance across many domain relevant factors is an application of multi-dimensional trust. Often an agent cannot satisfactorily accomplish a task alone; it must receive assistance, or in this example choose a partner agent. A technique for teaming agents was originally developed in [31] for solving the partner selection problem, and it serves as the foundation of expertise modeling for a recommender-based QA system in this research.

One substantial difference from trust algorithms as presented here and expertise modeling lies in the feedback mechanism. In basic trust, positive interactions result in increased trustworthiness assessments [41]. Interactions between agents are simply classified as a positive, negative, or neutral experience, and a single continuous value for trust is adjusted accordingly. With multi-dimensional trust, the updating mechanism is based on goal achievement, which is a composite of a number of properties, such as interaction quality, timeliness, et cetera. Like trust, an expertise model evolves with experience, but this evolution is guided by feedback concerning the quality of the content, which in a QA system is comprised of both questions and answers. This feedback is an evaluation of the worthiness or correctness of content within the scope of the topic(s) it concerns and colored by the expertise model of the evaluator. This feedback mechanism is further described in chapter 4.

2.2.3 Trust in Social Networks

Social networks can be explicitly defined using friendship or linking mechanisms, or they can be implicitly created, by users simply tracking the identity of a content creator. User generated content (UGC) websites often encourage users to formalize these networks of trust by providing various social networking features (contacts, follows, and friendships). These features are designed to increase engagement and loyalty in this active user base and to encourage the growth of the base in the long term. Due to these reasons, UGC forums are also under online social networks (OSNs) as *knowledge-sharing oriented online social networks*[32, 54] or content oriented social networks (COSNs). In the three primary activities users perform on OSNs, authoring content, viewing content, and networking, COSNs are OSNs where the emphasis is on authoring and viewing content. This is in contrast to *networking oriented social networks* (NOSNs) such as Facebook, which are driven by the users' social relationships and networking activity. Thus, on a networking oriented OSN, users will be most interested in information about their close friends, while in a knowledge oriented OSN, a piece of information may have intrinsic value (depending on its quality, relevance, etc.), independent of which member introduces it to the group.

Often such UGC sites rely on a small set of highly loyal and productive users whose actions interest the broader audience. Such users are the most trustworthy users. Historically trust is defined as a measure of the truthfulness or reliability of an agent [41]. In this research the most trustworthy agents are the users whose contributions to the community add the most value. The social networks on these sites, while helpful in increasing user engagement and allowing core users to quickly find information from sources they trust, can be problematic. The formation of social networks can give rise to various social phenomena such as nepotism, reciprocity, and cyber-balkanization [80], which can distort the rating processes of the core set of users.

One source of trust, apart from personal experience, is reputation. Reputation is an aggregate indicator of the trustworthiness of an agent as observed by other agents. A good reputation score implies that an agent, or user, is generally believed to be trustworthy. Barber and Kim explore this process of belief revision based on this type of reputation in [7]. Online communities, particularly general UGC websites, often have a large, sparsely connected user base. The likelihood of one user A interacting with another particular user B in a large system is very small, and often multiple interactions are necessary to develop an accurate model of direct trust. Therefore it is impractical to rely on direct interaction for a large part of their user base. Instead, such websites rely on an aggregated reputation model from the community as a whole, or "neighborhood reputation" [72], to identify valuable contributors.

Lerman *et al.* have investigated the spread of content in Digg and Twitter and discovered that the most prolific users find and consume content through their social networks [52]. These users also provide the most trust rating for reputation aggregation, hence their behavior patterns can be extremely influential. Lerman's work highlights the importance of social influence in governing which content is promoted in UGC websites. Similar findings have caused Digg to implement a policy where they discount endorsements, “Diggs”, by users who are in the same social network as the original poster [70]. We argue that while social influence causes users to vote for their connections, the opposite effect of homophily-based selection [19] needs to be taken into account as well. It is possible that users add others to their social networks because they like the content that they produce. In that case, discounting all votes from a user’s social network connections could be misleading. Instead, we propose an approach based on estimating the user’s intent behind his/her votes, instead of simply discounting all social contact based votes. In [27], Ghosh and Lerman show that it is possible to predict which content will flourish by examining the flow of that content through the social network in its early stages. Alternatively, this research focuses on identifying valuable contributors.

Building and trusting in others on an anonymous internet is difficult. Often there is little consequence for antisocial behavior, and users behave in a greedy manner. According to Resnick *et al.* effective trust and reputation models require that entities are long-lived, feedback about current interactions is captured and distributed, and past feedback guides buyer decisions [69]. UGC websites have the necessary infrastructure to address these points and build meaningful reputation models. Users on UGC websites have a persistent identity (their user names), and trust is established over time by observing their actions. Another fortunate benefit of building trust models of users in

UGC communities is the centralized nature of UGC websites. In contrast to decentralized trust models as proposed by Yolum and Singh [88], the website infrastructure of a UGC community monitors and aggregates every interaction. Constructing reputation in this centralized fashion allows all users to access the same reputation information for guiding their decisions.

According to Pavlov *et al.*, a potential pitfall of reputation information is that it may be provided in a strategic manner for numerous reasons including reciprocation and retaliation [65]. Social networks suffer from a similar problem. Phenomena between users such as nepotism, reciprocity, and retaliation can distort common measures of trustworthiness. To address this problem, this research describes a mixture-model based approach that explicitly models the behavioral aspects of interactions on a COSN as a component. The other component of the mixture is expected to model the process by which users assign unbiased ratings to high quality content. By estimating, for each user, the likelihood of their behavior belonging to either component, the algorithm attempts to identify users who indulge least in behavioral patterns that may mislead a reputation system.

2.3 Recommendation

One purpose of building expertise models is to perform recommendation. Recommendation is the process of matching users to other users or content. In the context of QA systems, recommendation is the process of suggesting open questions to an appropriate responder. A straightforward recommender-based QA system design would assume that a question is supplied, and the QA system must select potential responders from whom to solicit responses. This selection process, or responder recommendation, is a significant feature of the QA framework developed in this research. Recommendation is a challenging and popular problem in the data mining and machine learning communities, to the extent that Netflix offered a US$1M prize for an improved movie recommender [9].

Identification of expertise is the first step in recommending a responder. Expertise is defined as the ability of a user to answer a given question to the satisfaction of the questioner. Given a question, how can the pool of potential answerers be indexed and searched to predict who is capable of and willing to provide an answer. Estimating which user is most likely to give a satisfactory answer is a challenging problem and requires a complex model of human expertise along: a) expertise dimensions: the various distinct areas of human knowledge and the expert's ability in each of these areas, b) compatibility: the likelihood that the answerer's personality and approach to answering questions matches that of the questioner, c) willingness: the probability that the answerer will be willing to invest the time required to answer the question. The expertise model is further defined in chapter 4.

Multi-dimensional models of expertise are crucial to the problem, as an expert on one subject is not necessarily an expert on another, even when the two areas are closely related. While question-answer systems already exist on the web such as Yahoo! Answers and the defunct Google Answers, in most current implementations, there is only rudimentary modeling of user expertise, and hence an expert is expected to wade through many questions until finding one that is most suitable [14] [39]. Very specialized questions may never be viewed by the few qualified experts, and the result is that only simple and generic questions receive an answer. By automating the process of finding an expert for a question, we remove the investment of time required by an expert to find a suitable question, thereby reducing the cost of participation and improve overall productivity.

Expertise discovery is fundamental to recommending the best responders. There are two sources of information from which to model expertise: content-based and link-based [3]. Content-based expertise modeling analyzes the word usage by responders in order to build a model of the responders' expertise. This has been done in the context of clustering documents into topics by Griffiths and Steyvers [71]. Additionally, Zhang et al. have developed the QuME algorithm, which identifies expertise based on matching keywords [90]. Link-based expertise identification methods rely on evaluating the link structure between questioners and responders. Zhang et al. tested a number of network based ranking algorithms on data from the Java Forum, showing that link information can be used to identify expertise nearly as well as human raters [89]. Jurczyk and Agichtein applied a variation of the HITS algorithm to the much broader Yahoo! Answers data set and were able to identify authoritative users, or expert responders [42][46]. The expertise model in this work leverages both content and link information in order to best capture users' expertise in a QA forum.

A number of different modeling and recommendation techniques have been developed, but their application to QA has been limited. The GroupLens research group has developed a movie recommender based on information filtering and collaborative filtering agents [30]. Like the modeling approach in this research, their technique combines content and user analysis to make recommendations. Recommendation often blurs the line between different fields of research. While this research focuses on developing ways of selecting the most appropriate people to answer a question, others use recommendation to select answers from a large corpus of existing information, a process called information extraction [59]. This is very closely related, but instead of evaluating the fitness of a user, the fitness of a piece of arbitrary text must be evaluated. Like the proposed system, Andersen et al. have investigated trust-based recommendation systems and have developed a technique for determining incentive compatibility based on fundamental axioms [5]. These examples of recommenders illustrate the breadth of technology that can be applied to QA recommendation.

2.4 Incentive Mechanisms

While expertise models are useful for recommending questions and inducing suggested behavior, the incentive mechanism is used for rewarding actual observed beneficial behavior. These two techniques function as a push and a pull serving as separate means toward the same end. An incentive mechanism is an algorithm that must encourage optimal system-wide behavior from self-interested agents. According to David Parkes [64]:

> The mechanism design problem is to implement an optimal system-wide solution to a decentralized optimization problem with self-interested agents with private information about their preferences for different outcomes.

The proposed incentive mechanism is a collaborative aggregative system. An aggregative system is based on explicitly rewarding certain behaviors, while a collaborative system distributes rewards based on interactions between users. The collaborative aggregative incentive mechanism combines the simplicity and transparency of an aggregative system like that used in Yahoo! Answers with the flexibility and power of a collaborative mechanism, like the HITS algorithm [46]. Certain behaviors are rewarded, much like in Table 2.1, but rewards are calculated based on answer feedback, which is weighted by the influence of the reviewer.

The most interesting QA systems utilize incentives that bridge the gap between content oriented social networks (COSN) and networking oriented social networks (NOSN) in order to expand their user base and retention. Quora leverages real-world identities and reputation to attract users [20]. Users login with their real name and are encouraged to provide professional experience as an explicit listing of their credentials. This has lead to famous users from the tech world ensuring an audience within Quora based on their real world reputations. These users carry authority and act as a magnet to attract more people. Aardvark leverages social obligation in QA. Because people have an existing personal relationship with others in their network, when they are recommended a question to answer they are obliged to answer it out of a sense of social obligation. The responders know their work is going to directly help someone from their social circle.

Regardless of why people are motivated to participate in online communities, the problem remains of evaluating the importance or significance of one in a crowd. Search engines face this exact task, and Google uses a variant of the PageRank algorithm [63]. In a rapidly evolving graph like a QA system the normalized PageRank algorithm may be more appropriate for determining the value of a participant [10]. Kleinberg's HITS algorithm can be used to identify hubs and authorities in a network based solely on the link structure [46]. In addition to algorithms that are designed to calculate the importance of network content based on connections, an incentive mechanism can be built around techniques for ranking tournament participants [76]. Simply put, the process of selecting the most valuable users and rewarding them accordingly can be generalized across many domains. While the QA recommender relies on creating accurate models based on trust, the incentive mechanism relies on analyzing feedback concerning user actions.

Chapter 3

Q&A System Architecture

This chapter presents a novel architecture for questions and answer (QA) systems based on recommendation of questions to potential responders and a feedback driven incentive mechanism. A QA system is a prototypical online community, and this serves as a concrete application domain for the technology developed throughout this dissertation. Of the current major QA systems on the internet, only Aardvark attempts recommendation [1], and none have a non-trivial incentive mechanism based on non-monetary systemic rewards.

3.1 Architecture - User Perspective

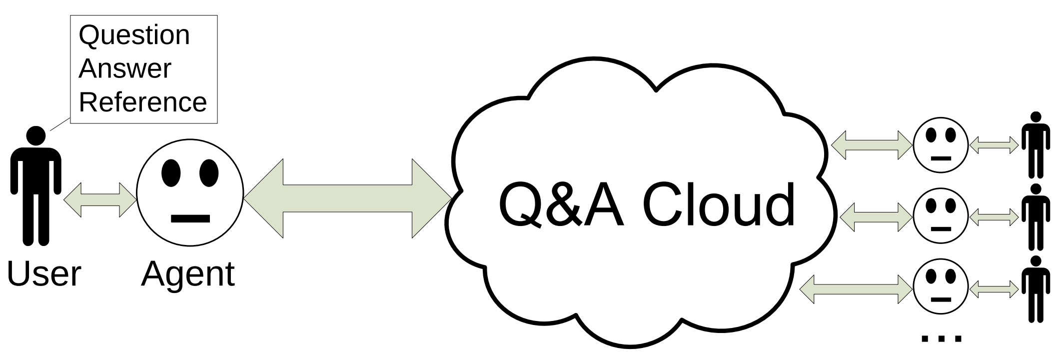

Figure 3.1: Q&A architecture, user perspective

Figure 3.1 shows the proposed architecture of a QA system from the user perspective. Each human user has his or her own personal recommender agent that automatically bridges the gap between users. A person must decide whether to ask a new question, answer an existing open question, or search the existing answered questions as a reference. The agent consults the QA cloud and retrieves the appropriate content for the user. Not only does the agent find appropriate content for the user's expertise, it retrieves open questions for answering based on a utility function that maximizes the expected reward earnings. This is further explained in Figure 3.2. Not shown on this diagram is the option for a user to evaluate the contributions of others. A user can endorse or denounce any content posted by another user, including both questions and answers. This user interface architecture represents a significant departure from the prototypical QA system, which uses a bulletin board style interface with a large listing of loosely related questions. This recommendation paradigm is designed to reduce the amount of effort needed to search for a question to answer based on question topic and potential reward. Along with a tangible reward, this reduced effort is expected to yield higher participation.

3.2 Architecture - Agent Perspective

Figure 3.2 shows a more detailed view of the proposed QA system from the perspective of a recommender agent. The agent is directly connected to the user and is capable of leveraging every bit of content produced or consumed by the user in order to make a user model. For the sake of scalability,

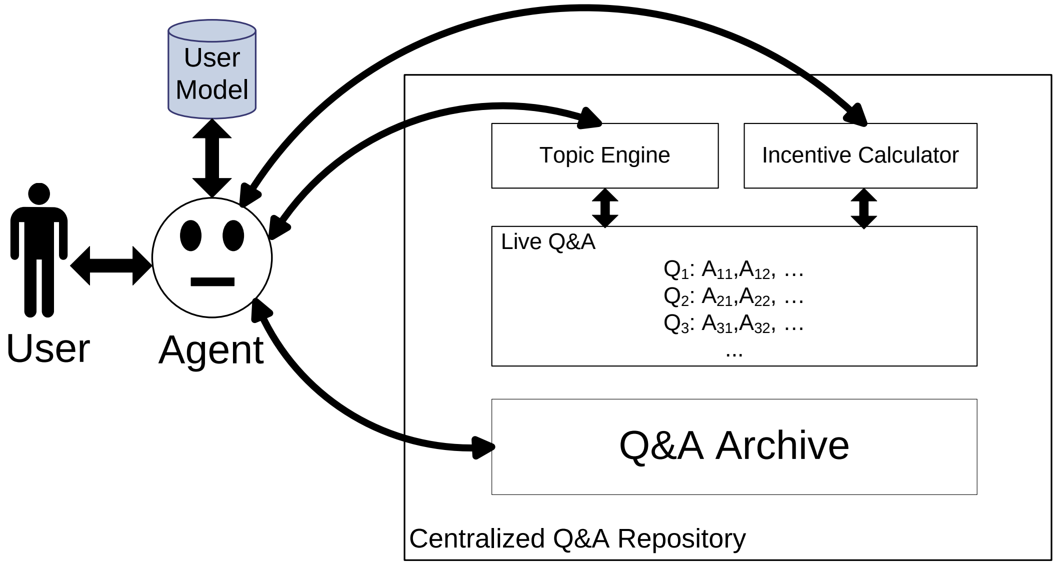

Figure 3.2: Q&A architecture, agent perspective

each agent models only its own assigned user. Competition is introduced when multiple users answer the same question. This is especially likely, and even desirable, when high value questions are asked. Rather than each agent modeling the capabilities of every user, the behaviors of outside users are captured in a probability that the assigned user will correctly answer a question. This is based on the topic of the question and the historical performance of the assigned user in this topic. The topic distributions of questions and answers alike are calculated by the Topic Engine, which is discussed further in section 4.1. The Incentive Calculator determines the amount of reward, or influence, a user receives based on the value of a question, analysis of feedback concerning responses, and user participation metrics. The mechanism that drives this Incentive Calculator is further discussed in section 5.

The recommendation agent is privy to the inner workings of the QA cloud. The simplest query is a reference request, or a search of the existing QA archive. Upon receiving this request, the agent will feed the query into the Topic Engine to generate topic metadata, search the archive, and retrieve content based on the words of the query along with the topic metadata. In addition, the agent will log the topic of the request in the user model. The purpose of logging reference requests is to capture what is interesting to the user.

Handling a new question from the user is slightly more complicated. The agent feeds the question through the Topic Engine, which calculates the posterior probability of the question belonging to each topic given the words of the question. This forms a probability mass function of topic membership for the question, meaning the question belongs to topic A with a probability x and topic B with a probability of y and so on. This concept of topic membership is discussed in detail in section 4.1. In addition, the Incentive Calculator calculates a question value based on the influence of the questioner. This question, along with its topic membership meta-data and value metadata is stored in the live QA database for others to answer. The design of the Incentive Calculator is discussed in section 5.

This proposed architecture is designed to be very accommodating to a user who wishes to respond to the questions of others. The user indicates to the agent that he would like to answer a question, and the agent retrieves questions that: 1) match the expertise of the user and 2) offer the highest expected rewards for the user. This is done by consulting the user model to find topics of expertise, then searching the live QA database for open questions that match this expertise. The resulting questions are sorted in order of expected earnings, and they are presented to the user. This saves the user's time by automatically suggesting questions that are targeted to the user's expertise instead of requiring a user to search through many pages of questions until finding an appropriate one. Moreover, this recommendation will serve to maximize the expected reward of a user, thereby increasing his influence in the system.

3.3 Architecture - Agent Decision Process

Figure 3.3 is a diagram showing the decision flow of the recommendation agent. At the top of the diagram is the human user who is connected to the agent. The shaded gray components are stages where the agent awaits input from the user. As in Figure 3.1, the user must specify whether he intends to ask a question, answer a question, or search for a reference question.

If a user wishes to reference a closed question, which is simply a question which is no longer accepting new answers, the agent must analyze the query for topic membership, search the QA archive, rank the resulting questions and answers according to a match of interest and relevancy, retrieve them for the user, and update the user model to reflect interest in the topic of the query. This is shown in Figure 3.3 in the decision path where `Query Type=Reference`.

When a user asks a question, the agent must send the question to the

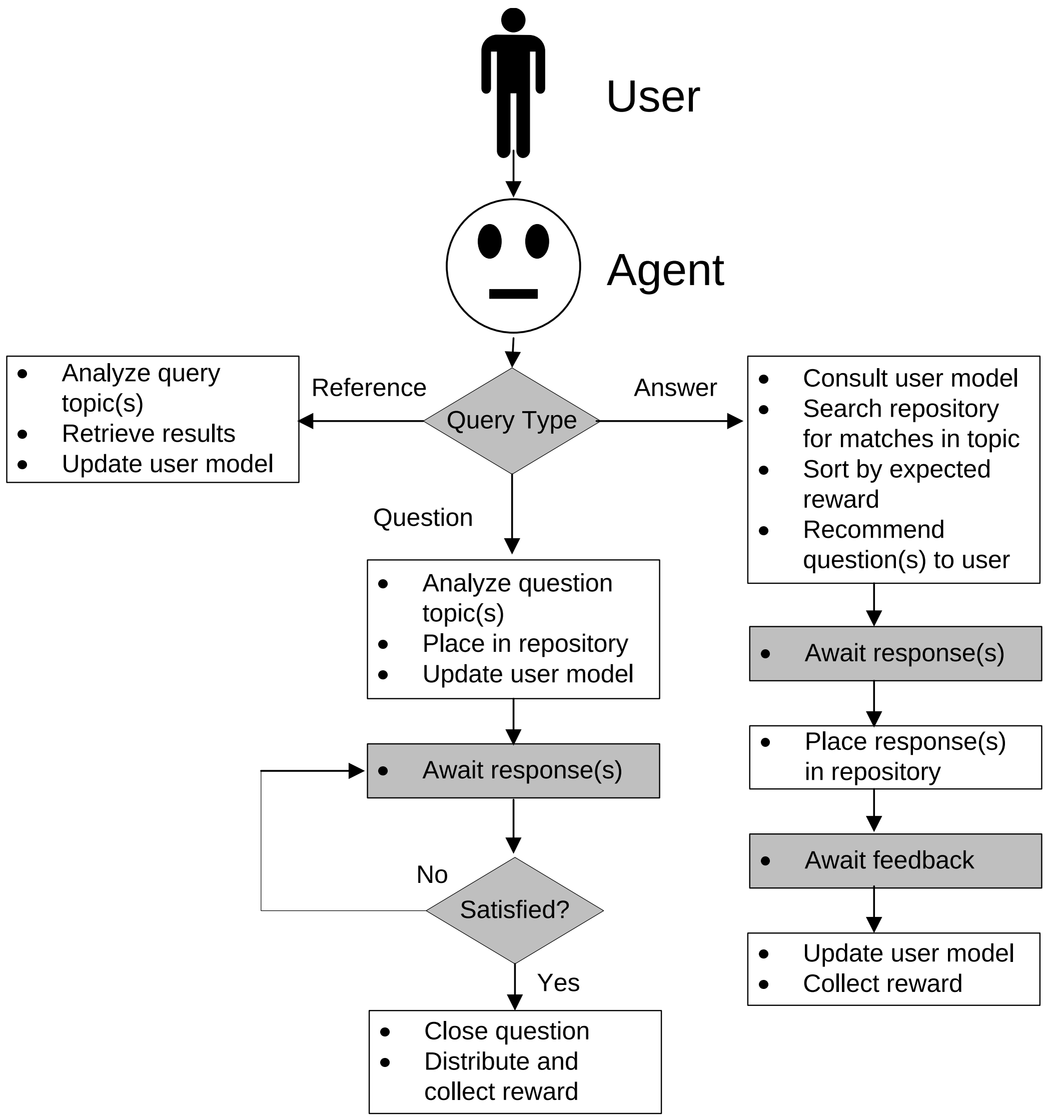

Figure 3.3: Recommendation agent decision flow

Topic Engine, which determines topic membership, and the Incentive Calculator, which determines the value of the question to a responder based on questioner influence. This is shown in the decision path where `Query Type = Question`. This question is then placed in the live QA repository while awaiting responders. The user model for the questioner is then updated to reflect interest in the topic of the question. Once other users have responded to a question, the questioner has the opportunity to give feedback on the answers. If the questioner is satisfied with one or more of the answers given, the question is closed, and the rewards are calculated and distributed to the appropriate responder(s). Also, the questioner is eligible to collect a small reward once satisfied in order to encourage question closing. It is important to note that the incentive mechanism is not a zero-sum game. That is, a questioner does not spend his influence reward by rewarding others. This lack of influence spending is designed to encourage an active community of participation, not inhibit users from asking questions.

When a user chooses to answer a question, the agent must take several steps as shown in the decision path where `Query Type = Answer` in Figure 3.3. As explained above, the agent must first consult the user model, then retrieve open questions that match the user's topic(s) of expertise, and present them in order of expected reward. Once the user has responded, the agent places the responses in the repository to await feedback from both the original questioner and from other viewers. The user can then collect the earned reward when feedback is given.

This proposed architecture is designed to encourage participation and foster user satisfaction in an online community by recommending appropriate questions to answer and rewarding socially beneficial user behaviors. The following chapters, 4 and 5, introduce techniques for evaluating trustworthiness and expertise, making QA recommendations, and creating and distributing participation incentives. These technologies govern the inner workings of the Topic Engine and the Incentive Calculator. The architecture presented here is given as an example of how these technologies can be combined to create more productive online communities.

Chapter 4

Identifying Experts and Recommending Content

This chapter introduces techniques for identifying experts in a question and answer (QA) system and evaluating the depth of expertise, or trustworthiness, of users in content oriented social networks (COSNs). Section 4.1 describes the technical foundation of a topic-specific models of expertise in QA systems. This section 4.1 describes a method originally developed by Budalakoti, DeAngelis, and Barber in [15] for modeling expertise and creating QA recommendations. This is presented along with results demonstrating the effectiveness of making recommendations on a static data set. Section 4.2 explains how to evaluate the trustworthiness or expertise of a user in a COSN, specifically in a news aggregation online community. This section describes and measures the performance of an algorithm for unbiased trust estimation on COSNs developed by Budalakoti, DeAngelis, and Barber in [16]. This is a difficult problem because social effects often overshadow the true value of content created by a user. This section introduces an algorithm for evaluating trust that is not biased by social effects.

4.1 Expertise Modeling and Recommendation

The Topic Engine in Figure 3.2 is responsible for two tasks: expertise modeling and question recommendation. Expertise modeling is a technique for calculating the posterior probability of a body of text belonging to each topic of expertise given the words in the text. In other words, the Topic Engine can analyze text and classify it into abstract groupings of subject matter. This is based on a higher level of abstraction called topics, which are distributions over words. This classification of text enables the identification of users' areas of expertise and facilitates matching between content and users, or recommendation. The technology behind this Topic Engine is an unsupervised learning algorithm. Unsupervised learning means the individual topics are not specified as in Yahoo! Answers, e.g. pets or sports [39]; topics are *discovered*. This means that topics are created organically, with no human intervention, allowing topics to evolve with the changing climate of the QA system.

In addition to expertise modeling, the Topic Engine is used for question recommendation. An effective QA system must facilitate the sharing of information. A question is provided by a user called a questioner, and any other user (called a responder) is capable of reading this question and providing a response. In this dissertation research a question is supplied, and the QA system must select potential responders from whom to solicit responses. This selection process, or responder recommendation, is a significant contribution of the proposed QA framework.

4.1.1 Topic Engine Innovations

The Topic Engine contains two primary innovations that are required for recommending questions to the users with the most appropriate expertise:

1. Expertise Model: Expertise is modeled in a QA system using higher level concepts called *expertise topics*, which are associated with distributions over words as well as users. These distributions are used to calculate the probability of an author having relevant expertise given a question, and these probabilities are used to make a recommendation according to a decision-theory framework. This method captures the historical word usage as well as participation patterns for users. This dissertation proposes a finite mixture based generative model to discover topics, and estimate the parameters of the model using the expectation-maximization (EM) algorithm.

2. Load Balancing Framework: A decision-theoretic framework has been created for recommending expert participants, which tries to find satisfactory responses for all questions without overloading any expert with too many questions.

An alternative to the generative model discussed in 4.1.3.1 is to use a standard clustering algorithm such as the k-means algorithm and identify these clusters as topics. An even simpler alternative is to forgo the explicit definition of expertise topics. This is accomplished by associating words with authors based on historical usage and then recommending author responders based on the similarity between the words in the question and the responders' word usage. This dissertation research intends to show that author-topic and word-topic models for describing QA behavior lead to better responder recommendations than a traditional scheme of clustering words and authors into topics or simple author-word counts. Experimental support for the generative model approach to expertise modeling is given in section 4.1.7.

The proposed recommender currently adopts the questioner perspective and recommends the most appropriate expert responders to answer a given question. It is possible to apply this same recommendation mechanism to recommend a question of the appropriate topic to an idle responder. This second approach to recommendation is the most appropriate way to match open questions to available responders as shown in the user perspective architecture in Figure 3.1.

A second experiment supporting the recommender design demonstrates the effectiveness of applying the proposed decision-theoretic framework to the responder recommendation problem. This experiment is also presented in section 4.1.7. Without the framework, it is possible to simply assign a question to a topic and choose the most appropriate author according to the author-topic distributions. This is called the *best match* technique. It is possible that the best match technique will lead to undesirable system properties such as the overloading of the most qualified experts with too many questions. The decision-theoretic framework is designed to balance the responder load and the satisfaction of the questioners when recommending questions, and this experiment will investigate its performance according to system-wide metrics.

4.1.2 Characterizing Recommendation

The proposed QA recommendation task addressed by the agent in the architecture figure 3.2 can be described as follows: given a potential responder, identify open questions concerning topics that pertain to the responder's expertise.

Verification of the modeling and recommendation algorithms has been done using static data, which requires an analogous form of recommendation: given a question by a user on a QA forum, identify users on the forum that are most capable of providing a satisfactory answer to the question. Previous historical information about interactions on the QA forum is required to train the system.

More specifically, this research assumes that the training data is available to the Topic Engine in the form of individual question-answers (QAs), and for each QA, the following information is available: a) a unique questioner id, b) text of the question, c) ids of all responders, d) the text of each responder's response, and e) some information indicating which answers were found satisfactory by the questioner. Then, in the test step, the identity of the questioner and the text of the question are given. The task is to suggest suitable responders for the question. Then the actual responders are revealed, along with the text of their answers.

In the case of a live test, it is possible to directly contact the suggested responders to judge their interest in the question. However, current system tests are performed using historical data derived from QA forums. In this case, recommendation quality is judged by how many of the recommended responders had originally answered the question.

4.1.3 Expertise Modeling for Recommendation

Most approaches to the recommendation problem can be divided into two categories: a) content-based filtering, and b) collaborative filtering. In content-based filtering, users are modeled based on what they have been interested in, or liked, in the past. In collaborative filtering, new recommendations are made to a user based on the interests of other users identified as most similar to them. In the QA forum setup, each responder supplies content in the form of text of the questions he chose to answer, as well as the text used to answer these questions. Also included is collaborative information about the other responders that chose to answer the same question as the responder, as well as the questioner.

A simple content-based approach to the expert recommendation problem would be to build a text-based profile, for example a term frequency inverse document frequency (TFIDF) profile, for each responder. Then, when the system receives a new question, the question text could be compared to all the responder profiles using a similarity measure such as the cosine score, and the responders with most similar profiles could be recommended the question. This is a common approach with respect to expertise modeling and recommendation, taken by Zhang [90] and Godil [29].

However this simple approach, which treats the expertise identification problem as a document retrieval problem, suffers from a serious drawback: an expert is very different from a document in the sense the exact words used by a responder are heavily contingent on the questions he/she chose to answer, and do not cover all the information that a responder has, or the topic he/she may be knowledgeable about. For this reason a simple text profile based approach is not sufficient for the purpose of modeling human expertise.

To overcome this drawback, this research introduces an alternate approach to expertise modeling, which models user expertise in terms of *topics*, instead of words. A topic can be seen as a higher-level concept over words, and is modeled as a distribution over words. Hence, two questions may belong to the same topic even though they may have no words in common. Similarly, an expert may be recommended a question even though there is no match in terms of profile words, if the question is judged as belonging to a topic the expert is interested in. Two words that are synonymous will have similar distributions over topics for two reasons. First, they are likely to co-occur in the same document (question and corresponding answers). Secondly, they are likely to co-occur with many of the same words.

The next section introduces a generative model for a collection of question-answers in a QA system. Learning the parameters of this generative model enables the identification of topics of interest to various authors, as well as topic-word distributions.

4.1.3.1 Generative Model for a Q&A Dataset

A generative model originally developed by Budalakoti, DeAngelis, and Barber in [15] is the heart of the expertise modeling algorithm that drives the Topic Engine. The generative model is a way to characterize the formation of questions and answers using statistical distributions. A basic outline of a generative model for any online forum where people might gather for a discussion or to exchange information can be constructed as follows: at each timestep, a topic, which might have its own prior distribution, is generated from a distribution over topics. Then, some (question) words are generated related to the topic. The topic distribution may or may not be independent of the original author of the post, depending on how closely people stick to their topic(s) of interest. Following this, a set of responders is chosen from a distribution, based on the topic, and each of these responders generates further words. The words generated by the responders are related to the topic but may or may not be seen as drawn from the same distribution as the topic. For example, users may have strong personal opinions, or try to draw the discussion in some favored direction. This might require modeling each user as having its own word distribution for each topic, or the words as drawn from a mixture distribution of the original topic distribution, and a word distribution related to the user.

The model outlined above will be expensive to create due to the large number of parameters involved. The model is simplified considerably, reducing it to a finite mixture model in the process. These simplifications reduce the number of parameters considerably, while still providing important insight into the dataset. The generative model is described below:

Assume that the number of unique words p, the number of topics $|T|$, and the number of unique responders s is known in advance. Let the words be labeled $1, \ldots, p$ and the users $1, \ldots, s$, arbitrarily. At each timestep, a topic t is generated from a multinomial distribution τ over topics, and a vector $\vec{w^q} = \{w_1, \ldots, w_p\}$ is generated, where w_i, is the count of word labeled i in the generated words. The words are generated from ϕ_t, a multinomial distribution over words corresponding to topic t. Following this a responder vector $\vec{x} = \{x_1, \ldots, x_s\}$ is generated from θ_t, a multinomial distribution over users for topic t, where x_i is the number of times the user labeled i responded. Each of the users in x in turn generates words based on the topic t. Here, a simplifying assumption is made that the words generated by a responder as part of the answer are drawn from the same distribution ϕ_t as the topic.

This assumption can be understood as saying that the words used in the answer to a question by a responder depend only on the topic of the question, and do not depend on any attributes of the responder. This is a reasonable assumption in QA forums where factual information is exchanged for the most part, or even in forums where personal opinions are expressed but the vocabulary used does not differ very much from user to user. It may not hold true in forums such as blogs or discussion forums, where responses to topics are much longer and more personal, and people may have favorite topics they might try to steer the topic toward. However, this level of model

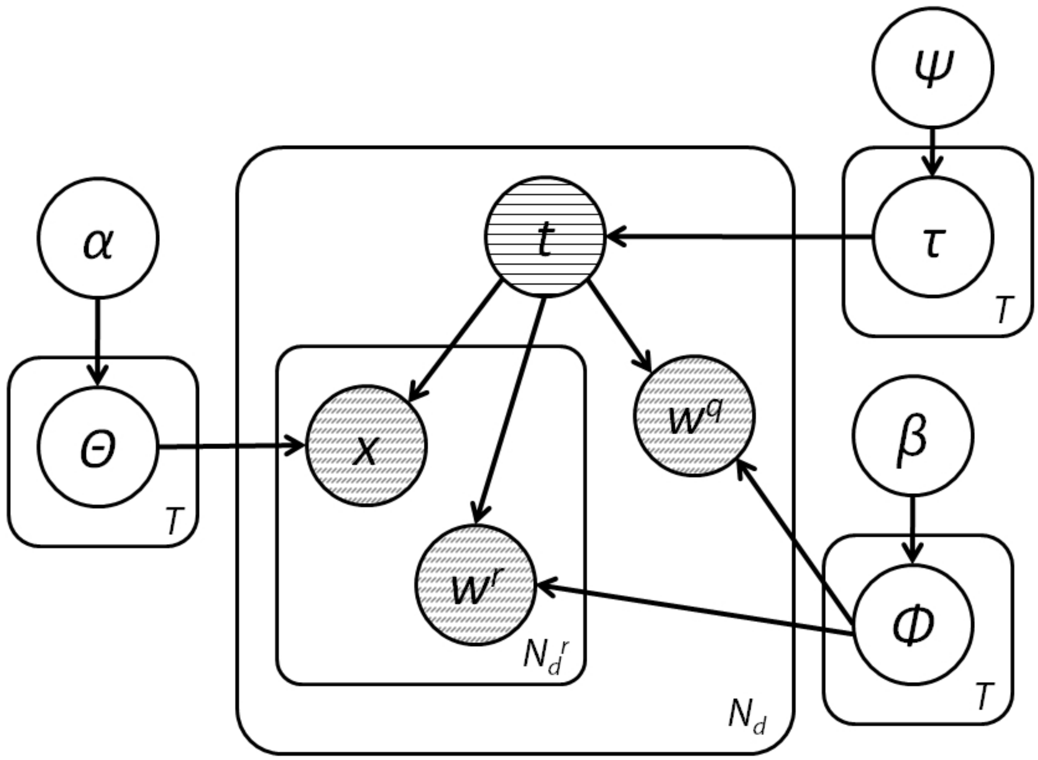

Figure 4.1: Generative model for a question answer recommender [15]

complexity is unlikely to be necessary for QA forums.

Figure 4.1 displays the generative model described above in plate notation. The shaded variables are the observed variables, while the unshaded variables are the hidden variables. The horizontally striped circle represents the topic, which is unobservable. Also, α, β and γ are symmetric Dirichlet priors, used for smoothing. Set $\alpha = \beta = \gamma = 1$. The total number of words and users generated for each question is assumed to be independent of τ, θ and ϕ, and hence their randomness can be ignored in our discussion. Also, since it is assumed that the words generated by the responders depend solely on the topic, the words generated by all responders can be written as a vector $\vec{w^r} = \{w_1, \ldots, w_p\}$. Therefore $\vec{w} = \vec{w^q} + \vec{w^r}$.

4.1.3.2 Learning Model Parameters

Figure 4.1 shows that the generative model is essentially a mixture model with a finite number of components, where the number of components is the number of expected topics in the dataset. Let the number of such components/topics be g. Let the total number of unique QA interactions in the dataset D be n, where the j^{th} such interaction is referred to as d_j. Then there is associated with each d_j a hidden vector z_j of length g, where $z_{ij} = 1$ if d_j is about topic i. Therefore $\vec{\theta} = \{\theta_1, \ldots, \theta_g\}$, $\vec{\phi} = \{\phi_1, \ldots, \phi_g\}$, and $\epsilon = (\bar{\theta}, \bar{\phi})$. Then, assuming z_j for all d_j, the log likelihood of ϵ given D is:

$$\log_D L(\epsilon) = \sum_{i=1}^{g} \sum_{j=1}^{n} z_{ij} \{\log \tau_i + \log P(\vec{w_j}|\theta_i) + \log{(P(\vec{x_j}|\phi_i)}\} \tag{4.1}$$

The expectation maximization (EM) algorithm [23] is used to to estimate the hidden variables z_i, and the parameters τ and ϵ. The derivation of the EM algorithm for the model is fairly straightforward. The Expectation (E) and Maximization (M) steps are given below:

E-Step: Given a guess for τ and ϵ, the expected value of z_{ij} is given by:

$$z_{ij} = \frac{\tau_i \cdot P(\vec{w_j}, \vec{x_j}|\epsilon_i)}{\sum_{h=1}^{g} \tau_h \cdot P(\vec{w_j}, \vec{x_j}|\epsilon_h)}$$

M-step: Given expected values of z_j, we can estimate τ, θ and ϕ as follows:

$$\tau_i = \sum_{j=1}^{n} \frac{\psi + z_{ij}}{\psi|T| + n}$$

$$\theta_{ik} = \frac{\alpha + \sum_{j=1}^{n} z_{ij} x_{jk}}{\alpha s + \sum_{j=1}^{n} \sum_{k'=1}^{s} z_{ij} x_{jk'}}$$

$$\phi_{ik} = \frac{\beta + \sum_{j=1}^{n} z_{ij} w_{jk}}{\beta p + \sum_{j=1}^{n} \sum_{k'=1}^{p} z_{ij} w_{jk'}}$$

4.1.4 Evaluation Metrics

Any evaluation of QA recommender systems needs to take into account the experience of users from both perspectives: as questioners and as responders. As questioners, users would like highly satisfactory responses. As responders, users would like not to be overloaded with too many questions, or be recommended questions they are not interested in. In particular, if they are high quality experts, there is a risk that they might be recommended too many questions in a bid to provide satisfactory responses to questioners, which might result in reduced participation from them.

Two new metrics, *responder load* and *questioner satisfaction*, measure the quality of QA recommenders from both of these perspectives. Both of these metrics are variations of *precision* and *recall*, metrics commonly used in information retrieval [75] and recommender system research [74].

It is possible to define precision and recall for a user in a QA system from two perspectives: as a questioner and as a responder. Therefore, four

Table 4.1: Metrics table for responder x [15]

	Answered Question	Did not Answer Question
Recommended	R_x^+	N_x^+
Not Recommended	R_x^-	N_x^-

Table 4.2: Metrics table for questioner x [15]

	Upvoted Answer	Ignored Answer
Recommended	U_x^+	I_x^+
Not Recommended	U_x^-	I_x^-

metrics of interest can be defined. However, the action governing the quality of all four metrics is the same: each time a new question is introduced in the system, the recommender makes a decision to contact a subset of responders in the system and recommend the question to them. The decision to answer the question is made by each individual responder and cannot be controlled by the recommender. The decision made by the recommender to recommend a question has to take into account both the possible impact on questioner metrics as well as responder metrics.

4.1.4.1 Responder Precision and Recall

Let all the participants in the QA forum be represented by X. Any $x \in X$ can be a questioner or a responder/answerer. Let the user x's responder precision be written as π_x^a, and responder recall as ρ_x^a. Then,

$$\pi_x^a = \frac{R_x^+}{R_x^+ + N_x^+} \qquad\qquad \rho_x^a = \frac{R_x^+}{R_x^+ + R_x^-}$$

Here, the right-hand-side terms are as defined in Table 4.1. Table 4.1 can be understood as follows: R_x implies that the responder x liked the question, i.e., responded to the question. N_x means that the responder x did not like the question. A superscript of $+$ means the recommender system suggested the question to the responder x. A superscript of $-$ suggests that it did not suggest the question to the responder. For example, R_x^+ is a count of the number of questions that are both liked by responder x and recommended to x.

For a given responder, responder precision is the ratio of the number of questions that were recommended to a responder, that the responder answered, to the total number of questions recommended to the responder. This is an important measure of the quality of the recommender, as a recommender that recommends too many irrelevant questions will drive away responders.

The recall for a responder measures what fraction of the questions the responder answered were recommended by the system. It is a measure of how well the recommender covers all the interests of the questioner, and while still important, it is relatively less significant.

4.1.4.2 Questioner Precision and Recall

Similarly, let the user x's questioner precision be written as π_x^q, and questioner recall as ρ_x^q. Then,

$$\pi_x^q = \frac{U_x^+}{U_x^+ + I_x^+} \qquad \rho_x^q = \frac{U_x^+}{U_x^+ + U_x^-}$$

Here the right-hand-terms are as defined in Table 4.2.

Questioner precision measures how many of the responders recommended by the system provided satisfactory answers. This is also relatively unimportant: a user will not usually mind extra answers so long as he is receiving a sufficient number of answers that are satisfactory. There may be problems in extreme cases, such as when a particular user is spammed, but this problem might be handled in other ways, such as allowing questioners to ban specific responders from their questions, or rank answers based on responder quality, or responder history.

Questioner recall measures how many of the answers/responders of interest to the questioner the recommender was able to identify in advance. This metric is more important than questioner precision because it indicates the degree to which questioners can depend on the recommender to find the most suited responders for a question.

Responder Precision and Questioner Recall are the focus of the recommender system tests. The next section presents some variant definition of Responder Precision and Questioner Recall. These re-descriptions make these metrics more intuitive to understand in terms of recommender system behavior.

4.1.4.3 Responder Load and Questioner Satisfaction

Define *Responder Load* λ_x for user x as $\lambda_x = 1 - \pi_x^a$, or:

$$\lambda_x = \frac{N_x^+}{R_x^+ + N_x^+} \tag{4.2}$$

The higher the value of λ_x, the greater the number of questions a responder has to read through to find questions of interest. In that sense, it is a measure of the load on the responder.

Define *Questioner Satisfaction* as $\sigma_x = \rho^q_x$, or:

$$\sigma_x = \frac{U^+_x}{U^+_x + U^-_x} \tag{4.3}$$

σ_x measures what fraction of the answers founds satisfactory by a questioner were from responders contacted by the recommender. It can be seen as a measure of how satisfied questioners will be with the QA system if the responders relied entirely on the recommender to provide them with interesting questions.

The quality of a QA recommender could be measured as $\sum_{x \in X} w_x \lambda_x$, and $\sum_{x \in X} w_x \sigma_x$. w_x could be set based on some individual user preferences for load and satisfaction, or set as $\frac{1}{|X|}$, to get the average values. In the experiments presented here, there is a tradeoff between λ and σ. For example, by recommending all responders in the system, the satisfaction σ_x is fixed to 1 for all questioners. However, this will have an adverse impact on the questioner load, as most of the questions suggested to a responder, he will not find interesting. Hence, the basic challenge in the problem is to provide quality answers to questioners without overloading the better experts among responders and to manage this tradeoff in a reasonable way.

4.1.5 Expected Utility

A composite utility metric manages the tradeoff between λ and σ in a principled manner.

$$U = \sum_{x \in X} w_x((\bar{\lambda} R_x^+ - N_x^+) + (\bar{\sigma} U_x^+ - U_x^-)) \tag{4.4}$$

The utility function expresses the tradeoffs $\bar{\lambda}$ and $\bar{\sigma}$, which can be seen as the tradeoff between responders and questioners. For example, setting $\bar{\lambda}$ to 10 would suggest that the system is willing to tolerate 10 incorrect recommendations for a single correct recommendation. In a system where experts are paid, high values of $\bar{\lambda}$ and $\bar{\sigma}$ might be acceptable. On the other hand, in a voluntary system where experts are generally busy, lower values of $\bar{\lambda}$ and $\bar{\sigma}$ might be a good idea. Two parameters $\bar{\lambda}$ and $\bar{\sigma}$ are used instead of one, in order to effectively manage the tradeoff between questioners and responders. This is discussed further in the next section.

4.1.6 Utility Based Recommendation

In this research, utility is a measure of the value of recommending a particular user, or potential responder, relative to other users given a question. A recommendation or a lack thereof is determined by calculating the expected change in utility for that recommendation. Calculating the expected change in utility requires an estimate of the availability and expertise of a user.
Definitions:

Availability: For a given agent/responder x, define its availability in a topic t, v_x^t, as the probability that the agent/responder will answer any given question in topic t.

Expertise: For a given agent/responder x, define its expertise in a topic t, e_x^t, as the probability that the agent's/responder's answer will be rated as satisfactory, given that the responder answers the question. Assume that the rating provided by the questioner as satisfactory/unsatisfactory does not depend on the questioner and depends only on the quality of the answer.

Thus, for a given question from topic t, the probability that a user x will answer the question is v_x^t, and the probability that he/she will answer the question satisfactorily as $v_x^t \cdot e_x^t$.

Let there be a question asked by a questioner q. Then suppose the recommender contacts/recommends responder a. If a answers the question, R_a^+ increases by 1. Hence U_a increases by $\bar{\lambda}$. If a does not answer the question, N_a^+ increases by 1, and the overall utility of a, U_a, decreases by 1. Then the expected change in utility U_a for responder a, if he/she is recommended a question by user q, written as $\Delta E(U_a^{(q,a)})$, is given by:

$$\begin{aligned}\Delta E(U_a^{(q,a)}) &= \bar{\lambda} v_a - (1 - v_a) \\ &= (\bar{\lambda} + 1) v_a - 1\end{aligned}$$

Similarly, the expected change in utility for the questioner, $\Delta E(U_q^{(q,a)})$, can be calculated:

$$\Delta E(U_q^{(q,a)}) = \bar{\sigma} v_a e_a - (1 - v_a e_a)$$

$$= (\bar{\sigma} + 1)v_a e_a - 1$$

The overall expected change in utility can then be calculated as:

$$\Delta E(U^{(q,a)}) = w_a \cdot \Delta E(U_a^{(q,a)}) + w_q \cdot \Delta E(U_q^{(q,a)}) \tag{4.5}$$

Expanding this gives:

$$\Delta E(U^{(q,a)}) = w_a \cdot ((\bar{\lambda} + 1)v_a - 1) + w_q \cdot ((\bar{\sigma} + 1)v_a e_a - 1) \tag{4.6}$$

The proposed recommendation approach can then be summed up as follows:

1. Given a question by questioner q: for each candidate responder a, estimate v_a and e_a.
2. Calculate $\Delta E(U^{(q,a)})$ based on these estimates of v_a and e_a.
3. Recommend the question to all responders for whom $\Delta E(U^{(q,a)})$ is positive.

4.1.7 Modeling and Recommendation Experiments

Experiments originally presented in [15] demonstrate a working implementation of the recommender based QA framework. 5000 questions and their answers from each of three different subject categories in Yahoo! Answers have been crawled, parsed, and processed. These categories are Astronomy and Space, Books and Authors, and Wrestling. The collected data represents approximately one month of activity within each category. These categories were chosen to represent a broad spectrum of the type of content available on Yahoo! Answers. After the pages were stripped of html, stemming and stopping algorithms were applied to remove affixes and unimportant words.

The first experiment compares the effectiveness of three different methods of identifying and modeling expertise. The first method uses a basic information retrieval (IR) algorithm that calculates the cosine similarity between the words used by an author in historical QA data and the words contained in a question. This is included as a baseline measurement. The second method uses K-Means clustering [33] to group question-answer documents into clusters. Users are then assigned a probability weight in each cluster based on the fraction of questions answered in the cluster. Given a new question, users are recommended based on the normalized weighted sum of the similarity of the question to each cluster centroid, multiplied by the probability of response (availability) of each user in the cluster.

The third method uses the proposed generative model to discover the author-topic distributions. Given a new question, a probabilistic estimate is made of the topic of the question, and responders are recommended based on their marginalized probability of responding across all topics, using θ.

Expertise models are built on a training set of 4000 questions and answers. The remaining 1000 questions and answers form the test set. For each of these questions in the test set the decision-theory framework was applied to calculate an estimated change in utility for each potential responder. The values for w_a and w_q represent the historic prolificacy of the responder, or the normalized number of times that responder and answerer responded in the past. Responders that are then recommended have an estimated positive change in utility. The values for $\bar{\lambda}$ and $\bar{\sigma}$ are chosen to tune the recommender performance. These values are chosen to encourage a large number of recommendations and therefore high questioner satisfaction at the expense of responder load. The recommender performance is analyzed using the metrics of responder load and questioner satisfaction. One more simple metric called *weak satisfaction* indicates whether the recommender was able to select at least one of the authors who actually responded. $\bar{\lambda}$ and $\bar{\sigma}$ have been selected to fix *responder load* near 0.95. This value of responder load represents a moderately high responder sensitivity to being overloaded. With a fixed load, it is easier to compare the tradeoffs made when evaluating questioner satisfaction. Assume that all actual responses are satisfactory. Yahoo! Answers collects best-answer information that indicates the single most satisfactory answer. However, the unpredictability of the dataset makes recommending the single most satisfactory response difficult and also marking a single response as satisfactory does not imply that all other responses were unsatisfactory. Table 4.3 displays the experimental parameters and results of an experiment to examine the impact of expertise modeling on recommendation.

The second experiment compares the performance of a recommender based on a decision-theoretic framework versus a very simple recommender that makes recommendations based only on the strength of an expertise match, neglecting the tradeoff between load and satisfaction. Table 4.4 contains the

Table 4.3: Expertise modeling comparison [15]

Expertise Modeling Parameters		Recommender Performance		
Astronomy & Space		**Responder Load**	**Questioner Satisfaction**	**Weak Satisfaction**
Info Retrieval	$\lambda = 35$ $\bar{\sigma} = 20$	0.9485	0.1187	0.3570
Clustering	$\bar{\lambda} = 50$ $\bar{\sigma} = 20$ $K = 30$	0.9480	0.2682	0.6440
EM	$\bar{\lambda} = 250$ $\bar{\sigma} = 250$	0.9559	0.2390	0.6567
Books & Authors		**Responder Load**	**Questioner Satisfaction**	**Weak Satisfaction**
Info Retrieval	$\lambda = 15$ $\bar{\sigma} = 5$	0.9743	0.0186	0.1141
Clustering	$\bar{\lambda} = 35$ $\bar{\sigma} = 20$ $K = 15$	0.9700	0.0079	0.0531
EM	$\bar{\lambda} = 200$ $\bar{\sigma} = 300$	0.9701	0.0419	0.1722
Wrestling		**Responder Load**	**Questioner Satisfaction**	**Weak Satisfaction**
Info Retrieval	$\lambda = 25$ $\bar{\sigma} = 10$	0.9740	0.0472	0.2472
Clustering	$\bar{\lambda} = 75$ $\bar{\sigma} = 100$ $K = 30$	0.9748	0.2115	0.7050
EM	$\bar{\lambda} = 300$ $\bar{\sigma} = 200$	0.9752	0.0922	0.4424

Table 4.4: Recommendation algorithm comparison [15]

Astronomy & Space		Responder Load	Questioner Satisfaction	Weak Satisfaction
DT Framework (EM)	$\lambda = 250$ $\bar{\sigma} = 250$	0.9559	0.2390	0.6567
Best Match	Top 25	0.9467	0.2102	0.6046

results of this comparison. The *Best Match* algorithm simply recommends the top 25 responders according to the expertise information from the *EM* algorithm. This is done to simulate a naive recommendation system that does not use a decision-theoretic framework. These two experiments have been performed to verify the design of the expertise modeling and recommendation algorithms. These two algorithms serve in the Topic Engine as shown in Figure 3.2.

4.1.8 Recommendation Analysis

Yahoo! Answers is a noisy and unpredictable dataset. Over a set of 5000 questions representing one month of activity in one category, 11588 unique users participated. Of these users, 4890 responded to a question only once, and 4723 never responded to a single question, but only asked questions. In addition, many users leave the system after a short period of time. 2319 new users appeared in the test data set of 1000 questions.

Running an offline test of a recommender is very difficult. The results show that the proposed system was able to correctly recommend at least one responder more than half the time (weak questioner satisfaction) while maintaining a load of $\lambda < .95$. This means that of every 20 recommendations one user responded to the question. Given the bulletin board structure of YA, it is certain that responders rarely see every available question. In a live recommender test, a much higher percentage of responses from recommended responders is predicted because it can be assumed that the responder sees the question and knows it has been recommended based on his expertise. Also, with a live test it would be possible to measure the questioner's satisfaction with any given response, leading to a more accurate measure of the recommender performance regarding questioner satisfaction.

Table 4.3 contains the results of the first experiment. A *responder load* of 0.95 indicates that of 20 questions recommended to a user, he responded to one. This number is artificially high because this is a measurement using an offline test. The results show that the *clustering* and *EM* algorithms outperform the basic information retrieval algorithm according to questioner satisfaction in nearly every case. This evidence supports using more sophisticated techniques for discovering responder expertise. The *EM* and *clustering* algorithms performed very similarly across all three data sets. While recommendation was more successful with some data sets, the relative performance of the algorithms was preserved.

Table 4.4 compares two types of recommender systems. The first is

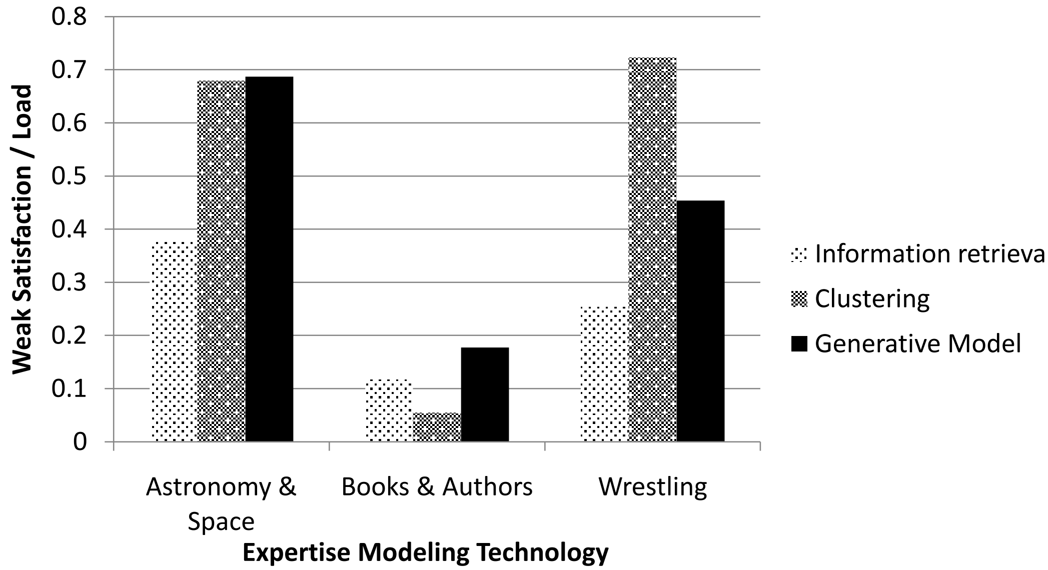

Figure 4.2: Expertise modeling technology performance [15]

the decision-theory based method, and the second is a simple *Best Match* algorithm. While this is just a single example, it shows that a more sophisticated utility-based recommender can outperform a simple recommender, even when they are supplied the same expertise information. The decision-theoretic framework only slightly outperformed the simple best-match recommendation algorithm. It is likely that interactive experiments run on a live dataset will better demonstrate the load balancing features of the decision-theoretic framework.

Figure 4.2 shows the performance of the three distinct expertise modeling technologies across all three harvested data sets from *Yahoo! Answers*. The performance metric is *weak satisfaction / load*, where *weak satisfaction* is fraction of times at least one satisfactory answer is recommended to a questioner and *load* is the fraction of irrelevant questions recommended to an expert, divided by the total number of questions recommended. A small *load* value and a large value for *weak satisfaction* are desirable, so higher values of *weak satisfaction / load* are preferred. From this figure it becomes clear that the more sophisticated techniques of topic-based clustering and generative modeling outperform the simple information retrieval approach. The two more advanced techniques performed comparably, but it is suspected that a richer data set will reveal additional performance gains by the generative model approach because this approach can leverage the power of author-topic and word-topic models instead of simply defining topics as clusters of words.

4.1.9 Recommendation Discussion

An expertise model and recommender for selecting the most appropriate responders given a question has been developed. This technology forms the core Topic Engine in a question and answer forum under development that is designed to encourage expert participation. The two primary components of this work are: 1) a finite mixture model based approach for characterizing the production of content in an online question and answer forum and 2) a decision-theoretic framework for recommending expert participants while maintaining questioner satisfaction and distributing responder load. The generative model uses word content information and collaborative information to build models of users' expertise that are employed during recommendation. Two new metrics have also been developed: responder load and questioner satisfaction. These metrics are used to evaluate the performance of the proposed recommender system on datasets harvested from Yahoo! Answers. Three methods of constructing expertise models are compared: a simple information retrieval approach, clustering words to discover their distribution over topics, and expectation maximization based on the generative model. In addition the decision-theoretic based recommender is compared to a simple best-match configuration. Experiments across several topic domains demonstrate the proposed system's ability to predict responder identities and suggest new responders.

These expertise modeling and recommendation techniques form the core of the Topic Engine in the proposed QA system architecture. These techniques allow the matching of questions to responders and consequently, responders to questions. This form of recommendation lowers the time investment needed by a user and bolsters questioners' confidence in the answers of others and therefore encourages more participation from the user base. The Incentive Calculator in Figure 3.2 also attempts to encourage participation, but it utilizes a system of non-monetary extrinsic rewards.

4.2 Unbiased Trust Evaluation in Social Networks

Online communities based on user generated content (UGC) often rely on a small set of highly loyal and productive users to identify or create content that would interest their broader audience. Through continual contact, this user base often develops informal reputational and social ties among themselves. Websites often encourage the formalization of these trust networks by providing various social networking features to increase engagement and loyalty. A user may come to trust a subset of the others as a consistent source of good content, while avoiding the remainder, leading to influence-based fragmentation. This section investigates the impact that these emergent social behaviors can have on the quality of reputation scores and develops algorithms that are able to take them into account while calculating user reputation. This section describes a method for evaluating trustworthiness that was originally developed by Budalakoti, DeAngelis, and Barber in [16].

4.2.1 Social Biases in Online Communities

The two key roles on a UGC forum (or COSN) are that of content creators, users who create content with the expectation that it may interest others, and users who consume the content. The two roles are not mutually exclusive; the same user may be a content creator or consumer at different times, depending on the circumstances. There are many ways in which consumers may express their opinion of a piece of content: the very fact that the consumer accessed the content (by say, clicking on a link to it) can be seen as a positive affirmation. Also, many websites provide ways by which users can express their approval, for example, by 'upvote' links for users to click. We call any such action by which a content consumer may express their approval of content as a *selection*.

A selection can be seen as a vote of confidence in the quality of content produced by the content creator, and the total count of selections can provide a good initial estimate of the quality of a user's content, or a user's reputation. One problem with this approach is that all users are not equally good judges of quality: some users may be more qualified, or they may be more involved in the forum, and thus have a better understanding of the goals and 'personality' of the forum. A more advanced approach would be to weight selections by some measure of the selecting users' reputation in the forum, an approach that might yield an algorithm similar to eigentrust [44], proposed for reputation estimation in peer-to-peer networks.

Another important problem is that, even in the case of users who may be highly reputed on a forum, the motivations behind a selection they made is not always clear. The reason for this is the social aspect of UGC forums: over time users develop social relationships with other users, and these relationships impact choices about the content they consume or favor. As UGC forums rely on these users to select content of general interest, incorporating these biases while identifying content of general interest can adversely affect the selection quality. Some documented examples of such biases are:

1. *Reciprocity*: A common social norm on many forums is for users to provide a reciprocal link in response to a link. This norm can be seen as a form of courtesy, but is also exploited by some users to increase their link count. Reciprocity of links is a well-documented phenomenon on the websites Flickr [50] and Twitter [85].

2. *Social Voting*: Many content-sharing sites such as Digg and Yahoo! Answers allow users to add other users as contacts or friends. The aim is to increase engagement: the site is designed so that users find it easy to get updates on the activities of their contacts. A side-effect is that since users find interesting stories via their contacts, users with many contacts find it much easier to promote their content. Social voting has been documented on the website Digg [27] as well as Flickr [53].

In other words, the reputation that users aggregate over time does not depend only on their quality, but also on many behavioral side-effects of their social network interactions. In this section, a mixture model is developed that assumes that a user's rating behavior could be driven by one of two intents/motivations:

1. *Content Quality*: The responder's expertise in a topic determines the quality of the content produced by him/her. A selection based on content quality recognizes the content creator's authority and should be considered when estimating his/her reputation.

2. *Social Affinity*: The social affinity between a content creator and producer is independent of the content quality and depends on their relationship with each other, which may be observed or modeled, given information about their online social network links.

4.2.2 User Reputation in COSNs

A natural way to define a user's reputation in a UGC forum is the number of times content created by him/her has been selected, or rated positively, by another user. A more sophisticated approach would be to weigh each selection by the reputation of the user making that selection.

Then, let the reputation (or authority) of user i in a topic be written as r_i and the number of times user j selected content by user i, r_{ji}. Let N_S be the total number of selections made. Then r_i can be written as follows:

$$r_i = \sum_{j=1}^{N} r_j \frac{r_{ji}}{N_S} \tag{4.7}$$

where N is the number of users. Now, let the number of ratings by user j be written as q_j. Then, after normalizing with the total reputation of all users in the system, we can rewrite r_i as follows:

$$r_i = \frac{\sum_{j=1}^{N} q_j \cdot r_j \cdot p_{ji}}{\sum_{j=1}^{N} q_j \cdot r_j} \tag{4.8}$$

where p_{ji} is the fraction of content by i selected by j. Dividing both numerator and denominator by N_S, we get:

$$r_i = \frac{\sum_{j=1}^{N} \frac{q_j}{N_S} \cdot r_j \cdot p_{ji}}{\sum_{j=1}^{N} \frac{q_j}{N_S} \cdot r_j} \tag{4.9}$$

Interpreting $\frac{q_j}{N_S}$ as the probability that user j will provide a rating, written as

P_j^q, we get:

$$r_i = \frac{\sum_{j=1}^{N} P_j^q \cdot r_j \cdot p_{ji}}{\sum_{j=1}^{N} P_j^q \cdot r_j} \tag{4.10}$$

4.2.2.1 Absorbing Random Walk Interpretation

Equation 4.10 can be written in matrix form: let Q be a diagonal matrix, where $Q(i,i) = P_i^q$, let P be a matrix such that $P(j,i) = p_{ji}$, and let r be a vector corresponding to $r_{i...N}$ above. Then the above equation can be written as:

$$(QP)^T r = r \tag{4.11}$$

Add a small uniform prior probability matrix ez^T to P, where $e_i = 1$ for all i, and z sums to 1. This signifies a small probability that any user can select any other user, even with no current evidence in the data. Adding 1 to the denominators of $Q(i,i)$ preserves a probabilistic interpretation. A restriction that r sum to 1 can be added. Then the above equation can be rewritten as:

$$(QP + ez^T)^T r = r \tag{4.12}$$

Solving this gives

$$r = (I - QP)^{-T} z \tag{4.13}$$

Let $T = QP$. Then[1] $r = (I - T)^{-T} z$. As all rows of matrix T sum to less than

[1] In practice, the inverse need not be calculated, but r can be calculated by solving the set of equations using Gaussian elimination, based on T and z.

1, T can be interpreted as the transition matrix for a reducible Markov chain with $N+1$ states by adding an extra recurrent absorbing state, which is the exit state. At any timestep, if the system is currently in state i, it transitions to the exit state with a probability $1-Q(i,i)$, and to another state j with probability $Q(i,i) \times P_{ij}$. Then $R = (I-T)^{-1}$ is the definition of fundamental matrix of an absorbing Markov chain, that is $R = I + \sum_{i=1}^{\infty} T^i$ [45]. So, if a random walk is executed across the absorbing chain, R_{ij} is the expected number of visits to state j before exit, if the walk started in state i. As z is a probability vector, $r = R^T z$ gives the expected time spent in each state, if the initialization probability of the walk at vertex i is given by z_i.

4.2.2.2 Relationship to Eigentrust

PageRank[13] is a popular algorithm for link analysis over a collection of hyperlinked documents. A variation of PageRank, called eigentrust [44], was proposed by Kamvar *et al.* to estimate user reputation in P2P networks. Applying the eigentrust formulation to the problem here would define user reputation as:

$$((1-c)P + cez^T)^T r = r' \tag{4.14}$$

where c is a parameter, called the teleportation probability, and usually set to 0.85. The common approach to solving this equation is via an iterative method. However, solving algebraically, as described in [34] gives:

$$\Rightarrow r' = (1-c)(I-cP)^{-T} z \tag{4.15}$$

Comparing equations (4.13) and (4.15), we see that r and r' differ only by a constant, $(1-c)$. So (4.13) provides a generalization of the PageRank vector: $r = r'$ when $Q(i,i) = c$ for all i, that is, when all users are weighted equally, irrespective of the number of ratings provided.

One intuitive interpretation of PageRank in the context of webpages is the random surfer model: intuitively, a webpage's authority is estimated as the probability that a random web surfer would visit the page given that he/she starts at a random page and selects a random outlink at each timestep. The vector r can be understood in terms of the random surfer model as follows: in PageRank, there is a constant probability c with which a surfer gets bored at any timestep and teleports to another random page. This seems reasonable for webpages, where the number of links provided may have little relationship to the quality of the page, but for UGC forums, more active raters are more likely to be seriously interested in the forum and likely to be better judges of content quality. In our formulation, the probability of random teleportation varies inversely with the number of ratings provided by the user. It would be useful to have this effect level off at some point, so that users cannot increase their influence as content creators simply by evaluating heaps of content. For this reason we use a sigmoid function to set Q. We set $Q(i,i) = \frac{1}{1+e^{-0.05q_j}}$. This means that for questioners who have provided 100 or more ratings, $Q(i,i)$ is effectively equal to 1.

4.2.3 Mixture Model Based Reputation Estimation

Raters' fairness or objectivity is supposed to estimate the degree to which their ratings are motivated by the quality of the content rated, as opposed to the influence their social network has on them. Define a hidden variable vector *objectivity* o, where o_i is a measure of the degree to which rater i is fair with the ratings he/she provides. For raters motivated by content quality, $o_i = 1$, and for raters completely driven by their social network, $o_i = 0$. We make the simplifying assumption that all of a rater's selections are driven by one of the two motivations. As part of the reputation estimation algorithm, we estimate the probability that $o_i = 1$ for each user i. To estimate o_i, we model a rater's behavior as follows: the number of ratings q_j each user provides is drawn from a distribution (this distribution need not be modeled as part of the final algorithm). Each user also has a hidden variable o_i associated with him/her. Following this, for q_j timesteps, depending on the value of o_i, the user i draws values from one of two distributions: the quality distribution (if $o_i = 1$) and his/her personal social affinity distribution (if $o_i = 0$.) Let O be a diagonal matrix where O_{ii} is the estimated objectivity value of user i.

The quality distribution ρ is defined as follows: the user selects another user at random, with the probability of user j being selected proportional to their reputation r_j. Essentially ρ is the same as the vector r, normalized. The social affinity distribution σ_i for user i is defined as the user's social network, with all members equally likely; people who are not members are assigned a small prior, to assure nonzero likelihood. We use another prior: the prior probability of selecting from the social affinity distribution defined for each user, which is the number of times the user selected a poster, or content creator, from his/her social network, based on historical data. We refer to this as the affinity prior π. Then given a set of selections, the posterior probability of selecting from either of the two distributions can be calculated. The quality distribution depends on O, as only users who are objective should be considered while calculating ρ. However, re-estimating ρ changes the objectivity values O for all users. An iterative expectation maximization based algorithm has been chosen, where user objectivity and the quality distribution are alternatively estimated.

4.2.4 Unbiased Trust Experimental Support

Experiments were conducted testing the trust algorithm on data from the user generated content website Digg[2]. Along with Reddit[3], Digg is currently one of the most used content aggregators. Digg maintains a rich, active user community and contains the necessary components for trust estimation in a content-oriented social network including: user generated content, a voting and aggregation system, and a mechanism to link users into a social network. Digg social network and endorsement data was obtained with permission from Lerman *et al.* [52].

The dataset represents one month of front page activity in 2009. For

[2]www.digg.com
[3]www.reddit.com

Step 1: *Initialization*

a) Initialize $\vec{\pi}_i$, the affinity prior probability for each user i.

b) Set $\rho = (I - OQP)^{-T}\vec{z}$, where Q and $\vec{z}$ are initialized as described in Section III.

c) Repeat Step 2 to 4.

Step 2: *Objectivity Estimation*

For each rater i in the dataset, and their ratings $\vec{s_i} = r_{ij}$.

estimate $o_i = \frac{(1-\pi_i)P(\vec{s_i}|\rho)}{(1-\pi_i)P(\vec{s_i}|\rho)+\pi_i P(\vec{s_i}|\sigma_i)}$.

Step 3: *Likelihood Estimation*

a) Calculate

$LL^{(j)} = \sum_{i=1}^{N}(1 - o_i)\log P(\vec{s_i}|\sigma_i) + o_i \log P(\vec{s_i}|\rho)$,

where j is the current iteration number.

b) If $LL^{(j)} < LL^{(j-1)}$, exit.

Step 4: *Reputation Estimation*

c) Set $\rho = (I - OQP)^{-T}\vec{z}$.

Figure 4.3: Iterative algorithm for reputation estimation [16]

each user submitted link (story) that made it to the front page we have access to the identity of the story poster and the identity of each user who 'diggs' the link. Additionally, for each of these users we have access to the single-directional link data, indicating that a user is 'following' another, thus forming a social network. Each user has access to the activity of the users he/she follows, so that when a user diggs a link, all users who follow him/her are able to see this information. A significant portion of the votes on Digg come from this process, where users find content that their friends have endorsed, a process described as a 'cascade effect' [52]. These endorsements are driven by a mixture of two classes of motivators: similarity-based and social influence-based. Similarity-based motivation occurs when a user follows a content creator because of a preference for content by that content creator, whereas social influence-based motivation occurs when a user endorses content from a creator because of a social relationship with that creator. Because these motivations are mixed, it is difficult to identify users who submit preferred content from those who are merely socially influential.

The aim of the experiments is to test whether a mixture model based approach that attempts to model social interaction dynamics can identify users relying unfairly on their social network influence to boost their reputation. This is compared to a PageRank [13] based approach that does not take into account any information about possible social motivations of voter endorsements (diggs). It is expected that the algorithm will identify users who provide better quality content. As a measure of content quality, we use the mean num-

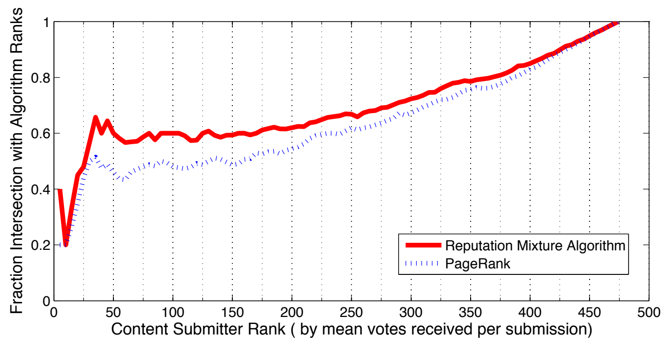

Figure 4.4: Fraction of user ranks predicted correctly by reputation algorithm and PageRank [16]

ber of votes received by a user once their story is promoted to the front page, because a large majority of votes for a front-page story come from the website's broader audience, making it difficult to rely on social affiliations. For the experiments, we analyze, for each content creator/poster, the voting data for each story they have posted until it receives 30 votes. This information is used to calculate the reputation of each user using our mixture-model based algorithm. The correlation of the reputation scores observed with the mean number of votes received per story for each poster can be calculated, and this is compared to a naïve PageRank based approach.

Table 4.5 shows the correlation coefficient values of the reputation and PageRank scores of each story submitter with the total votes received by his/her stories. The correlation is high in both cases but higher for the reputation algorithm. Table 4.6 compares the averaged reputation and PageRank scores (obtained by dividing reputation/PageRank scores with number of submissions) with the mean votes received per submission. This is a better

Table 4.5: Correlation: Reputation and PageRank scores vs submitter total votes [16]

	Correlation Coefficient
Reputation Mixture Model	0.895
PageRank	0.809

measure of a content creator's quality than the aggregate number of votes, as a user can be inconsistent in quality but still receive a large number of votes in total if he/she submits a large number of stories. However, in this case, the correlation is weaker. But the reputation algorithm still outperforms PageRank in correlation.

To compare how well the two algorithms rank users by quality, the scores provided by each of them are sorted in descending order and compared to a ranking of posters by mean number of votes received. The comparison is shown in Figure 4.4. The y-axis of the graph shows the fraction of users in common between the ranking of users by mean vote per submission and the ranking generated by the algorithm. The reputation algorithm identified two of the top five ranked contributors, while the PageRank algorithm could not identify any. However, both algorithms could identify only two of the top ten. This is responsible for the initial drop in performance of the reputation algorithm from a peak. Following this the reputation algorithm consistently outperforms PageRank.

Table 4.6: Correlation: Averaged reputation and PageRank scores vs submitter mean votes per post [16]

	Correlation Coefficient
Reputation Mixture Model	0.591
PageRank	0.484

4.2.5 Discussion

Content oriented social networks populated with user generated content are growing in popularity and diversity. Many such networks rely on large numbers of users who voluntarily generate content. This content draws in other users and creates value for the site. It is important to be able to identify the most valuable users and establish a level of trust in these users. This trust can be harnessed in the form of reputation, which is a signal that can be shared with others to drive decision making. Additionally, knowing the valuable members of a community is useful for the system designers because the designers can then implement strategies and incentive mechanisms to draw more trustworthy users to a site.

Social effects often hinder the performance of existing reputation mechanisms in UGC communities. This work presents an algorithm that is not biased by social effects and demonstrates the performance of modeling user reputation in a COSN. The performance of the algorithm on UGC data from Digg has been demonstrated, and the results are applicable to any content-oriented social network relying on user generated content. While this section is not specific to question and answer systems, this trust estimation technique can be applied to QA systems or any other community based on user generated content.

4.3 Chapter Summary

The techniques presented in this chapter allow one to identify experts and recommend content (section 4.1) in a question and answer system, and also to evaluate the trustworthiness and expertise of users in a more general context, regardless of social effects (section 4.2). These techniques form the foundation of a new class of online communities because they allow us to identify the most valuable contributors from a group of peers. Recall $RQ1$ in section 1.2.1:

> *RQ 1: How can the expertise of a user in an online community be measured?*

The techniques presented in this chapter directly address this question of measuring expertise in several types of online communities. Once these desired contributors are identified and characterized, it is possible to develop incentives to encourage their beneficial participation. Chapter 5 explains how such an incentive mechanism is constructed. Referring back to the example architecture presented in chapter 3, section 4.1 details the inner workings of the Topic Engine for building topic-specific models of expertise and recommending content, and chapter 5 discusses the functionality of the Incentive Calculator.

Chapter 5

Incentives For Online Communities

Chapter 5 describes the nature of incentive mechanisms, specifically in question and answer (QA) systems and describes the design of a novel incentive mechanism based on reciprocal systemic rewards. The Incentive Calculator in Figure 3.2 is responsible for encouraging participation by allocating influence points, or rewards. These points serve as a valuable reward due to the principle of reciprocity. The points themselves have no value, but a user possessing a large number of points will be given priority when asking a question. This forms a *systemic reward*, a reward that has value within in the context of the system. Users can earn more influence by satisfactorily answering questions from others with high influence. Game theory is relevant to the QA problem because users do not always answer questions to the best of their ability, but they are in competition for influence points, and they are typically rational actors. Mechanism design is a particular field of game theory, originally developed for applications to economics. These applications include auctions, markets, pricing strategies, and many others. Game theory traditionally focuses on careful observation and strategy surrounding the behavior of rational agents in a given competitive game. Mechanism design can be considered a form of reverse game theory; the game is designed to induce certain strategies and therefore certain outcomes. Like Parkes [64], Tuomas Sandholm defines mechanism design as:

> Mechanism design is the art of designing the rules of the game so that a desirable outcome is reached despite the fact that each agent acts in his own self-interest [73].

Each user, or rational agent, has a *type* (θ), which represents their full capabilities and expertise. From their contributions to the QA system, we can see their reported type, or $\hat{\theta}$. A social choice function maps the true type θ to an outcome, whereas a mechanism maps the reported type $\hat{\theta}$ to an outcome. The goal of mechanism design is to design a game that has an equilibrium state that implements the social choice function. Therefore, it is imperative to fully understand the desired outcome of a game before a mechanism can be created.

As stated in chapter 1, the desired outcome of the proposed QA system is to increase user participation and satisfaction. The recommendation-based QA architecture described in section 4.1 and the non-monetary extrinsic reward based incentive mechanism are two complementary, yet independent, technologies for pursuing this same goal. The most significant difference between these technologies is that expertise modeling and recommendation are driven by Bayesian models of user generated content, while the incentive mechanism is based solely on user actions and human feedback regarding original content. This distinction allows the automatic classification of user expertise with no human specification or evaluation needed. This automated expertise classification is capable of capturing dynamic expertise evolution as a user becomes more proficient in new topics. In contrast, it is desirable to harness human feedback for the allocation of influence points. This human evaluation leverages the wisdom of the crowd to solve the difficult task of evaluating responder content. Ultimately the entire purpose of a QA system is to provide answers that satisfy the original questioner and provide value to the rest of the community, and the best judges are people.

The purpose of an incentive mechanism is to pursue a desired outcome. In the proposed QA system, the desired outcome is increased expert participation. This high level outcome can be deconstructed into more manageable system goals. These are distinct from the goals of individuals. Individuals would like their own questions answered promptly and satisfactorily, with little regard for the questions of others. System goals concern the general welfare of all users in the system.

Many smaller goals can support the overarching target goal of increased expert participation and satisfaction stated in the hypothesis. Two primary classes of behaviors are desired. The first and most obvious behaviors are expertise based. It is certainly preferable to elicit responses (and questions) from users with a high level of expertise. Users who have more expertise in a topic are more likely to provide satisfactory answers in that topic. The second class of desirable behaviors concerns participation. Even if a user is not capable of providing deep insightful answers to particular questions, there is still value in contributing something. Imagine the case where an easy question has been asked. A user who frequently responds to questions on a very short notice but perhaps without the greatest expertise will be able to solve this question effectively and satisfy the questioner. The proposed QA system is designed around the assumption that people have varying levels and areas of expertise, but everyone has something valuable to contribute. Therefore reward, or influence, is based on both demonstrated expertise and participation. It is important to note that the proposed mechanism, while based on influence points, does not imply the creation of a market. Points cannot be *spent*. Asking a question, even one that is rated poorly, does not remove points. This design decision was made to encourage participation from all users, regardless of perceived expertise.

These goals are expressed in a set of desired *outcomes*. A full understanding of the desired outcome is necessary in order to develop the rules that form the incentive mechanism. The following is a list of desirable outcomes in a QA system:

- Users are not penalized for asking a question or giving a poor answer.
- Satisfactorily answering a question yields a greater reward than unsatisfactorily answering that question.
- Satisfactorily answering a question of higher value (asker has more influence) should yield higher rewards than answering a question of lower value.

- Users who answer very difficult questions should be rewarded for doing so.
- Recently added users should be able to earn a meaningful amount of influence in a reasonable time in order to compete with more established users.

Some of these desired outcomes may sound counterintuitive, particularly the first one. A user should not be penalized for asking a question because part of the value of a QA system is having a rich corpus of questions and answers readily accessible as a reference. Discouraging asking questions reduces value for the questioner who is seeking answers and also for users who would benefit from answering the question. Poor answers made in earnest should also not be discouraged. It is important to encourage participation, and the cost of ignoring or filtering poor answers in minimal.

This list of outcomes captures the desired behavior of users interacting on the proposed QA system. The difficulty with mechanism design is mapping these outcomes to a set of rules for distributing rewards that enforces these outcomes with self-interested agents. Essentially, the problem of designing an incentive mechanism can be expressed in three steps:

1. Identify a set of desired outcomes (expert participation)
2. Select a reward that is meaningful to the target audience
3. Distribute the reward in a fashion that maintains the desired outcomes

These steps correspond to the three research questions presented in section 1.2. Expert participation as a desired outcome requires a meaningful definition of expertise. Chapter 4 describes several ways identifying the target demographic of experts according to expertise topic and content creator reputation. The next section, 5.1, takes a closer look at what types of incentives would motivate the participation of live users in a QA based online community. This section provides a solution to research question 2. This is followed by the development and evaluation of a full incentive mechanism in sections 5.2 and 5.3 in response to research question 3.

5.1 Choosing an Incentive

Online communities and peer-production systems rely on human users to create, curate, and moderate content. Such communities add value to the users by maintaining and distributing this content. Often the content creators act purely out of goodwill and a sense of helpfulness.

It is increasingly common for online communities to use *achievement based* incentives to motivate users to participate. Such incentives include leader-board standings, custom titles, trophies, and avatars. This is essentially giving virtual prizes for participating in the online community. Such incentives are very effective for a portion of the population. The leading users on Yahoo! Answers will often answer 80+ questions per day, every day, for as long as the website has been live [39]. These fanatical users are strongly motivated by the achievement based incentives.

A *reciprocal incentive* is fundamentally different. A reciprocal incentive rewards people who answer questions in a QA website by assigning them a score, which is then used to calculate the reward that another person will receive when answering the first person's questions. Therefore, people who answer the questions of others will be given priority when asking questions of their own. This is like gaining priority access to the knowledge of the entire community in exchange for providing answers. The key difference is that the rewards in a reciprocal incentive mechanism have systemic value. These rewards directly help the recipient accomplish something within the system. An achievement reward must provide its own value in isolation to the recipient. Recall $RQ2$ in section 1.2.2:

> *RQ 2: How effective are systemic reciprocal rewards for encouraging expert participation in an online community?*

This research posits that a reciprocal incentive would encourage greater participation, particularly from experts, than an achievement incentive in an online community. A common form of online community is a question and answer forum, and this domain was chosen for further investigation. A required step in creating such a mechanism is to decide what reward to give beneficial users. In economics, the reward is typically monetary because it is valued by the vast majority of people. In popular websites this approach is rarely feasible [14]. In order to address this first question of what type of reward to give and to gauge human interest in reciprocal mechanisms, a survey was performed comparing achievement based incentives and reciprocal incentives in QA systems.

5.1.1 Survey on Q&A Rewards

A short web-based survey was administered to 380 anonymous volunteers, including mostly engineering graduate students. The survey was through the Google Docs platform as a shared document. No personally identifiable information was collected, and the test subjects were not compensated for their participation in any way. The survey link was emailed through several distribution channels, and participants were encouraged to spread the link to others. The vast majority of participants took the survey within three days of its posting. Therefore, there is some selection bias, as most of those who took the survey have much in common, and this does not accurately represent the user base of a general purpose question and answer website. It is also possible that respondents *think* may not be a very strong judge of what actually motivates themselves. A more conclusive test would involve constructing identical full QA systems that differ in only the type of incentive provided to participants. An A/B test could then be performed to compare the two incentives and check for significant differences in preference [57]. This is further discussed in section 5.3. Additionally, responses are available only for those who volunteered for the survey, and presumably these people may be those who are most outspoken against current practices in online communities.

With this caveat in place, it is important to stress that the purpose of this survey was simply to measure interest in alternative incentives, specifically reciprocal incentives, for participation in online communities. The survey presented to the volunteers is as follows:

Q&A Survey

Thank you for participating in this survey. The purpose of this survey is to gain insight into how people behave in online communities. The survey is completely anonymous and optional. The results will be used only in aggregate. Please complete the survey no more than once, but feel free to pass it to others. If you have any questions please contact me at deangelis@mail.utexas.edu.

1. What is your highest education level?
    - Some high school
    - High school diploma
    - College degree
    - Graduate/Professional Degree
2. Are you familiar with question and answer (Q&A) websites such as Yahoo! Answers, Stack Overflow, and Quora?
    - Yes
    - No
3. Have you used such sites to look up the answer to an existing question? This includes arriving at a Q&A site through a search engine such as Google.
    - Yes
    - No

4. Have you used such sites to ask a question?
   - Yes
   - No
5. Have you used such sites to answer a question?
   - Yes
   - No

Incentives for Participation The following questions examine two different classes of incentives for an online question and answer system. The two classes are 'achievement-based incentives' and 'reciprocal incentives'.

6. An 'achievement-based incentive' rewards participation in a Q&A website with leader board standings, custom titles, trophies, and avatars. This is like gaining virtual prizes for participating online. How much would an 'achievement-based incentive' motivate you to participate in an online Q&A website?
   - A lot
   - Some
   - A little
   - None at all
7. A 'reciprocal incentive' rewards people who answer questions in a Q&A website by assigning them a score, which is then used to calculate the reward that another person will receive

when answering the first person's questions. Therefore, people who answer the questions of others will be given priority when asking questions of their own. This is like gaining priority access to the knowledge of the entire community in exchange for providing answers. How much would a 'reciprocal incentive' motivate you to participate in an online Q&A website?

   - A lot
   - Some
   - A little
   - None at all

8. Would you be more likely to use a question and answer website featuring a 'reciprocal incentive' or one featuring an 'achievement-based incentive'?

   - Reciprocal Incentive
   - Achievement Incentive

5.1.2 Survey Results

Over a period of 3 days, 380 people responded to the survey. This section presents the results of this survey along with significance analysis.

Figure 5.1 shows the education level of the survey responders. As expected, the survey sample is comprised of mostly graduate students. The intuition behind this question is that those with more education may have greater expertise and the ability to contribute more to a QA system, and it could be useful to measure this. Practically, this question serves more as an

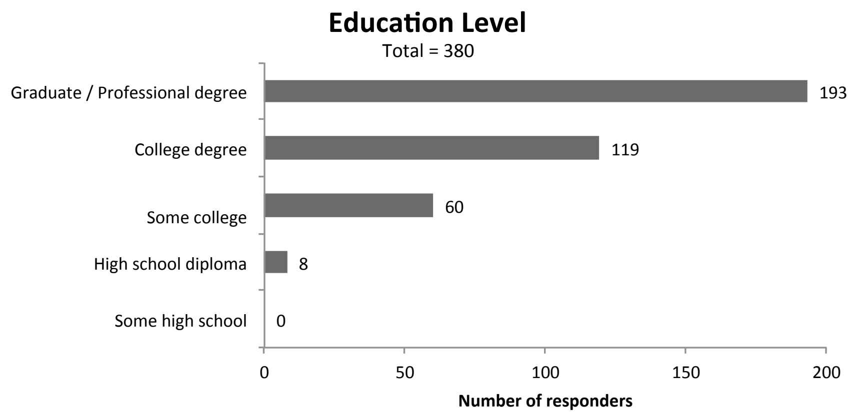

Figure 5.1: Survey responder education level

indicator of responder age and perhaps socio-economic status.

Figure 5.2 shows the level of experience that the responders have regarding online QA systems. Each of the four axes represents a yes/no question from the survey, and the position of the bold line as it crosses the axis indicates the number of affirmative responses. This figure shows that the vast majority of the survey responders are familiar with QA websites and gain value from them by looking up existing question and answer pairs. It is suspected that a large portion of the reference usage was driven by search engine query results. Only roughly one quarter of the responders have used QA systems in an active sense; that is, they have asked or answered questions. It is likely that this is due to a lack of suitable rewards for participation.

Figure 5.3 shows how the survey responders answered questions six and seven. This provides strong evidence that a reciprocal incentive is preferred to an achievement incentive. Differences shown for each of the bar pairings,

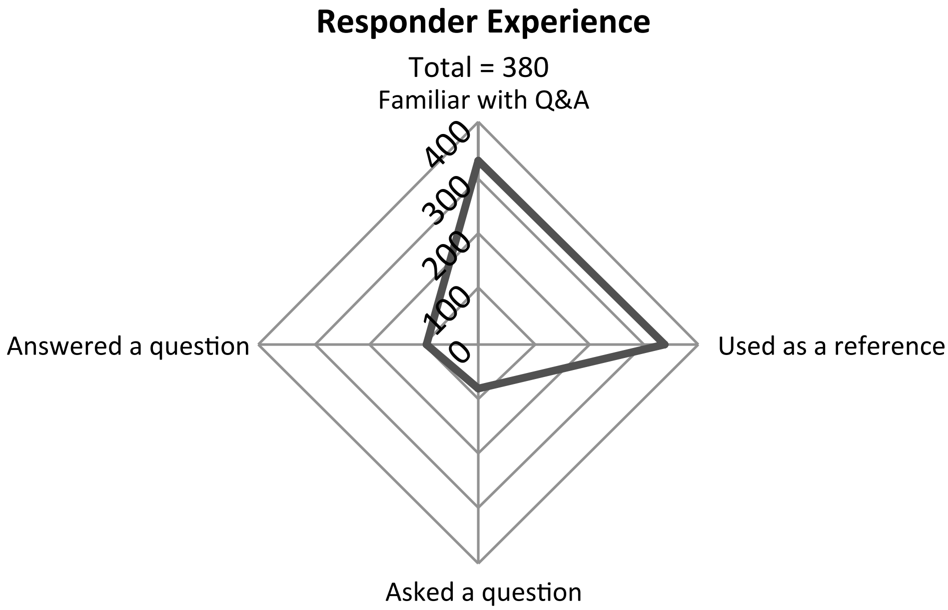

Figure 5.2: Survey responder Q&A experience level

except the bars representing the "a little" response, are statistically significant according a binomial test with $\alpha = 0.01$. This shows us that approximately 75% of the responders claim that an achievement incentive motivates them to participate only a little or none at all. In comparison this number for the reciprocal mechanism is only 55%. This point is made only more salient by looking at the other end of the spectrum. Nearly twice as many people rated the reciprocal incentives as having some or a lot of effect as compared to the achievement incentive.

Figure 5.4 shows the overall user preference between achievement incentives and reciprocal incentives. It becomes clear that reciprocal incentives are preferred by nearly twice as many survey responders. This question was

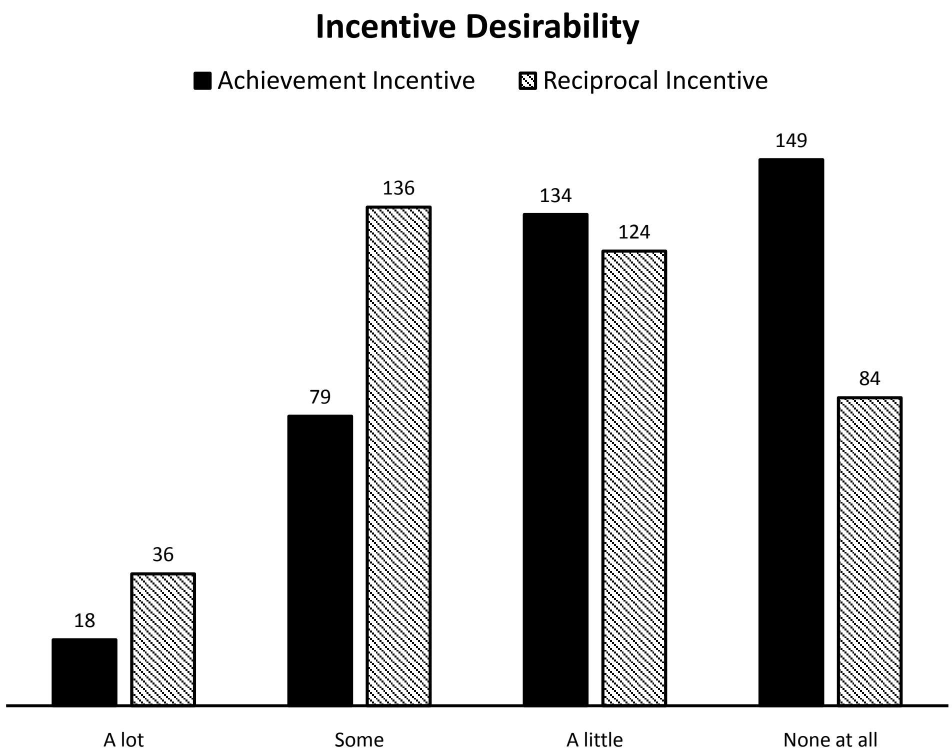

Figure 5.3: Survey responder claimed incentive efficacy

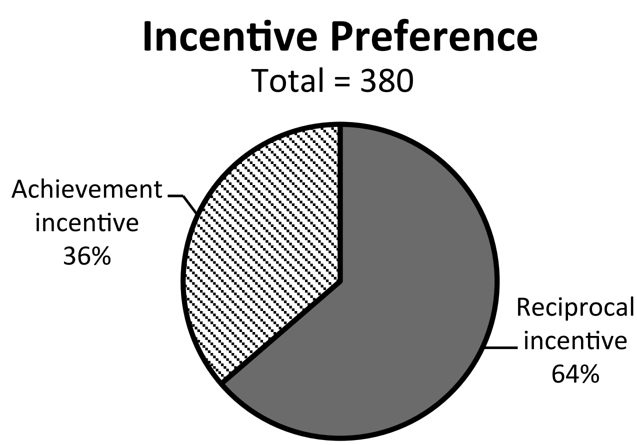

Figure 5.4: Survey responder incentive preference

Table 5.1: χ^2 test of independence

Variable 1	**Variable 2**	χ^2	***df***	***p-value***
Achievement	Reciprocal vs. Achievement	67.76	3	$< .0001$
Reciprocal	Reciprocal vs. Achievement	22.07	3	$< .0001$
Education	Reciprocal vs. Achievement	0.8	3	0.8495
Familiar with Q&A	Reciprocal vs. Achievement	0.01	1	0.9203
Used as a reference	Reciprocal vs. Achievement	0.07	1	0.7913
Asked a question	Reciprocal vs. Achievement	1.91	1	0.167
Answered a question	Reciprocal vs. Achievement	0.06	1	0.8065

intentionally written in a very unambiguous fashion, and the responders were required to choose one or the other, with no option to mark "undecided". This question does not capture the strength of the preference for reciprocal incentives, but it does support the use of such incentives. In practice, using a reciprocal incentive does not preclude using additional achievement-based incentives. Many successful online communities function very well with achievement incentives. This survey simply indicates that there is strong interest in creating something different. There is no reason that a community cannot be built using both reciprocal and achievement based incentives in order to appeal to the largest target population.

Table 5.1 shows the results of a χ^2 test for independence of the various factors in the survey[1]. The first column, *Variable 1*, represents a particular question from the survey, while *Variable 2* is the final question regarding overall preference. The χ^2 test for independence measures the independence of these two variables. In other words, "Is the final incentive preference (recip-

[1]Note that for *df=1* the χ^2 value reported is the Yates χ^2, corrected for continuity.

rocal vs. achievement) dependent on the answers to the question expressed in *Variable 1*?" The third column is the χ^2 statistic and the fourth column contains a measure of the degrees of freedom. The fifth column, *p-value*, represents the significance of the relationship. As expected, the variables *Achievement* and *Reciprocal* are not independent of final preference. This means that questions 6 and 7 regarding incentive effectiveness do affect the final preference of reciprocal or achievement incentives. This is expected behavior, because those who are motivated by achievement incentives would choose to prefer achievement incentives, and the same holds true for reciprocal incentives. This table becomes more interesting for the remaining tests. It indicates that preference for reciprocal or achievement incentives is completely independent of the other variables, including education level, familiarity with QA, reference usage, questioning usage, and answering usage. This means that reciprocal incentives are significantly preferred, regardless of how the dataset is sliced into subsets.

We have established that reciprocal incentives are preferred over achievement incentives in the survey presented here. There is a chance that the responders do not have an accurate picture of how these rewards are earned. Perhaps they underestimate the amount of effort needed to earn a valuable reward. At some point many easy to earn achievement rewards are preferred to a more time-consuming reciprocal reward. Nevertheless, this survey strongly indicates that reciprocal incentives are vastly more desirable than the current standard, achievement incentives. With this information we can construct an incentive mechanism based on reciprocal systemic rewards for encouraging expert participation in an online community.

5.2 Developing an Incentive Mechanism

Once a suitable reward is found, as described above in section 5.1, the next step is designing a mechanism to distribute the rewards in exchange for positive behavior in order to maximize the occurrence of the desired outcomes X as described in section 5. For the analysis in this section it is assumed that the reciprocal systemic rewards are desired and that the utility of these rewards is linear with respect to reward quantity. This section introduces notation and terminology originally developed by Hurwicz and Reiter in [36] for designing economic mechanisms. Here it has been adapted to QA systems. Question and answers systems can be considered privacy-preserving games of private information, or Bayesian games. Users, or agents, generally know their own expertise, but that is not necessarily public information, it is considered private. Users are not forced to answer questions or share their full knowledge, though they can choose to, hence the game is privacy-preserving. Each user, or agent, in a QA system is capable of answering questions honestly, promptly, and to the best of its ability. This optimal set of behaviors is called called the agent's true *type* and is represented with the symbol θ. A user is said to *report* its type by expressing certain behaviors. The observed actions of agents are then called the *reported type*, represented as $\hat{\theta}$. The set of all possible types, or behaviors, that a user can take, including asking questions, answering questions, evaluating content, and defecting from the system is called Θ. The mechanism y is a set of rules, or a function, that takes into account the game environment, g, and is executed on a reported type, $y(\hat{\theta})$. The result of this mechanism is an allocation of influence points, or a particular *outcome*, z, in the set of all outcomes Y. Therefore:

$$y(\hat{\theta}) : \Theta \rightarrow Y$$

The goal of mechanism design is to design an allocation y based on reported type $\hat{\theta}$ that has an equilibrium state ξ that implements the social choice function $f(\theta)$. The social choice function can be considered a target benchmark for the mechanism. $f(\theta)$ maps the true type θ, not the reported type $\hat{\theta}$, which can include deception or fraud, of each agent to the set of desired outcomes, X.

$$f(\theta) : \Theta \rightarrow X$$

The revelation principle states [61]:

> For any Bayesian Nash equilibrium there corresponds a Bayesian game with the same equilibrium outcome but in which players truthfully report type.

An incentive mechanism is designed to operate in a particular equilibrium state. At equilibrium, agents report their type as a function of their true type, $\hat{\theta}(\theta)$. Searching for equilibria in a Bayesian game is very difficult because the action space for each agent is large. It can choose to answer questions honestly, or it can lie, or it can refuse to respond at all. The revelation principle allows us to restrict our search to just those states where agents truthfully report their type, $\hat{\theta} = \theta$. In other words we must consider only states in which agents are honest, albeit selfish participants.

Consider the set of desired outcomes X in section 5 and consider the set of rules below:

1. Users earn influence points when answering questions correctly.
2. Answers deemed incorrect or spam do not receive a reward.
3. The influence points earned by a user answering a question are dependent on the influence of the user who asked the question.
4. An influence point bonus is awarded for authoring the best answer to a question, and this bonus is also dependent on the influence of the user who asked the question.
5. Users have a nonzero influence point balance when entering the system.
6. Influence points decay with time.

The first desired outcome is that users are not penalized for asking a question or giving a poor answer. There is no penalty for these behaviors specified in the rules. The only penalty is implicit. That is, users will waste their own time and effort by giving poor answers or asking worthless questions. Closely related to this is the second desired outcome that satisfactory answers are worth more than unsatisfactory answers. According to rule 2, unsatisfactory answers are not rewarded.

The third desired outcome is that answering a question of higher value yields higher rewards. Rule 3 supports this outcome because users who answer higher value questions correctly receive greater rewards than those who answer lower value questions. These higher value questions are those that originate from highly influential users.

Rule 4 enforces the fourth desired outcome that users should be rewarded for answering very difficult questions in place of ordinary questions. Very difficult questions do not necessarily have a larger reward associated with them because rewards are based on questioner influence, and more influential users do not necessarily ask more difficult questions [39]. This concern is why a "best answer" bonus is built into this mechanism. A more difficult question is likely to draw fewer answers, increasing the chances of giving the designated best answer and earning the bonus reward.

Rules 5 and 6 support the fifth desired outcome that recently added users should be able to compete with more established users. Returning users may build up substantial influence through participation over time. A new user is likely to have very little accumulated influence even though he may have significant expertise. This disparity means a new user's question is likely to have low priority, while the experienced user will be given high priority simply for having participated over a longer period of time. A decay function applied to a user's accumulated influence would ensure that only active members are given priority over others.

Proof. Let $I_t(q)$ be the influence of user q at time t

When $t = 0$,

$$I_t(a) >> I_t(b)$$

Because of Rule 5, No participation by a implies

$$\Rightarrow \lim_{t \to +\infty} I_t(a) = 0$$

Sustained participation by b implies

$$\Rightarrow \exists n (I_n(b) > I_n(a))$$

□

As described above, this mechanism maps the reported types $\hat{\theta}$ to the desired outcome, X. Because of the revelation principle, we can say that the mechanism implements the social choice function f. Each agent fares best when truthfully reporting their type, or participating to the fullest extent of their abilities, regardless of the actions of other agents. Therefore we can say the mechanism is incentive compatible. This creates a Bayes Nash equilibrium ξ where each agent reports truthfully and earns maximal rewards. There is no incentive for agents to deviate from their strategy of truthful reporting when others have not also done so. Moreover, the strategy that arrives at this equilibrium point is a dominant strategy. Regardless of the behavior of others, it is always in the best interest of a user to answer questions to the best of his or her ability.

This mechanism has been shown to be incentive compatible. Simply put, this means the mechanism encourages beneficial behaviors in individuals, while not encouraging damaging behavior. Incentive compatibility does not mean the mechanism is optimal however. There is perhaps a better mechanism for inducing the desired outcome. The optimal mechanism is domain specific. The optimal mechanism for one QA system may not be identical to the optimal mechanism for another. Optimality can be achieved through rigorous experimentation on a specific implementation and with very precise domain knowledge. Creating an optimal incentive mechanism for QA systems is outside scope of this research.

This mechanism is incentive compatible because it implements the social choice function when operating at a Bayes Nash equilibrium point where agents participate honestly and to the best of their abilities. However, there is one potential weaknesses: collusion. Collusion occurs when multiple users work together to exploit the system. For instance, if a user with a high level of influence creates meaningless questions and a second user responds to these questions while the first user rates the answers highly, the second user will gain rewards rapidly. However, relative differences in influence are meaningful. If many users have high levels of influence, the value (question priority) for any one of those users drops. Therefore, in smaller systems the mechanism does protect against this type of fraud. Additionally, users who are not in collusion can mark this content as spam, thereby eliminating the value to those in collusion. A more dangerous weakness is the threat of shared accounts. Multiple users operating under the same username are likely to have more expertise and availability than a single user. Therefore, it is likely that they will have a higher influence score. If many people band together under a single name each person would reap the rewards of a high influence score. Fortunately there are infrastructure-level ways to thwart this fraud. A simple example is disallowing a person to be logged in from two locations simultaneously. Section 5.4 contains a more thorough discussion of fraud in QA systems in general. The following section contains an experiment that compares the performance of this mechanism to the industry standard as implemented by Yahoo! Answers.

5.3 Testing the Incentive Mechanism

Chapter 4 is devoted to identifying the target audience of experts in an online community. Section 5.1 describes a type promising type of incentive, and section 5.2 describes the design of a mechanism for a question and answer system. With these pieces in place, it is possible to test the expected performance of a full QA system.

Ideally such tests would measure the expertise and participation levels of a population of users interacting on a live QA system. At one point on Yahoo! Answers there were approximately 120 million users and 400 million answers [51]. This yields a participation rate of roughly 3.3 responses per user over their entire lifespan on the system. The number of questions seen by each of these users, or the number of impressions, is unknown. Assume this number is 100. This means that Yahoo! Answers has a conversion rate of $0.0\overline{3}$. A 25% improvement on this performance requires a conversion rate of $0.041\overline{6}$. An A/B test for significance would then require $> 3,000$ impressions in the test group and $> 3,000$ impressions in the control group to show that the experimental group based on the new incentive mechanism outperforms the control group based on Yahoo! Answers with 95% significance [57]. If the measured improvement is $< 25\%$, then more impressions would be necessary.

Such a study would involve building a fully functional QA system, recruiting several thousand users, and randomly assigning them to control and experimental groups. Such a study is outside the scope of this research. A large commercial QA site expressed interest in running this experiment on their existing live user base, but negotiations proved unsuccessful. For these reasons a software simulation based on observed human behavior was constructed to compare the performance of the experimental incentive mechanism to the standard model, as used in Yahoo! Answers and modified by others.

5.3.1 Experimental Setup

A software simulator was created to compare an incentive mechanism based on reciprocal rewards to an incentive mechanism based on achievement rewards. This simulation was populated with agents designed to mimic human behavior in current QA sites, such as Yahoo! Answers and the Java Forum [89]. The Python programming language was chosen to implement the simulation.

The simulation begins by instantiating a fixed number of agents I, each representing a human user. These agents begin with a fixed number of reward points upon instantiation. The expertise x of each agent i is represented by a normal distribution. Two fixed numbers, the expertise mean x_{μ_i} and the expertise standard deviation x_{σ_i} are unique to each agent and used to define this distribution.

$$\forall i \in I, x_i \sim N(x_{\mu_i}, x_{\sigma_i})$$

Because it has been observed that users' abilities follow a power law distribution [89], the expertise means x_{μ_i} and expertise standard deviations x_{σ_i} are assigned according the following equation, where r is a uniformly random number in the range $[0.0, 1.0)$ and m and s are fixed constants. This expertise initialization matches the observed participants. There are exponentially fewer participants at the higher expertise levels.

$$\forall i \in I, x_{\mu_i} = r^m$$

$$\forall i \in I, x_{\sigma_i} = rs$$

Once these expertise models for each agent are initialized, a simulation cycle begins. One cycle is defined as a process in which:

- A random subset of the agents generates questions.
- Each agent has the opportunity to view some subset of the generated questions and estimates an expected reward.

- Each agent then ranks the questions it has seen in order of expected reward and chooses to answer a subset of these questions. This ranking is based on the expected reward calculation, which is dependent on which incentive mechanism is currently applied.

- Answers are generated and rewards are distributed based on the quality of the answer and the quality of others' answers.

- If the reciprocal incentive mechanism is in effect, then a decay factor is applied to the standing point balances for each agent.

- Some subset of the agents defect and leave the system, while some new agents are introduced.

Typically users with lower expertise are more likely to ask questions. This simulation models this as a linear relationship, where the probability of an agent asking a question in a single cycle $P(A_i)$ is defined below, where α is a constant, called the *question ask constant*.

$$\forall i \in I, P(A_i) = \frac{-x_{\mu_i} + 1}{\alpha}$$

The set of all questions q_j is called Q. A question q_j has a difficulty, d_j, which is defined below. Note that the difficulty of the question is *not* a function of the asker's expertise, x_{μ_i}. This matches observations that experts do not necessarily ask more difficult questions. The questions may simply be in a topic that the users have very little expertise in, see section 4.1. Less difficult questions are much more plentiful, however. Therefore this is also modeled as a power law distribution, where r is a uniformly random number in the range $[0.0, 1.0)$ and D is a fixed constant called the *difficulty exponent*. Also, under the control mechanism based on Yahoo! Answers the agent who asks a question has 5 points deducted from its balance. There is no deduction in the experimental reciprocal mechanism.

$$\forall q_j \in Q, d_j = r^D$$

Once the questions for that cycle are generated, the agents must select which questions to answer in order to maximize their reward. It is unrealistic that every agent can observe and calculate a predicted reward for every available question. This would be equivalent to a human reading the entire database of open questions on Yahoo! Answers, which numbers in the hundreds of thousands [39]. Therefore, the probability that any given question q_j is considered by agent i, is calculated as $P(C_{q_j,i})$.

$$[\forall i \in I, \forall q_j \in Q], P(C_{q_j,i}) = \frac{\beta}{|Q|^K}$$

$|Q|$ represents the number of questions, and β and K are constants called the *question seen constant*, and the *question exponent*, respectively. This equation indicates that as the number of questions grows, the probability of a single agent seeing one particular question shrinks exponentially. The simulator has additional functionality that can fix $P_{q_j,i} = 1$ for all q and j. This mode of operation emulates an ideal recommender. A recommender recommends content to users, and in the context of QA systems, it will recommend a question to a user who wishes to answer a question. An ideal recommender would examine all possible questions and return an optimal subset of questions to answer. Fixing the probability that a question is considered to 1 ensures that all possible questions are considered, and the agent can then select the questions to answer from the entire pool of questions. Section 4.1 describes recommendation in detail.

Let the set of all considered questions q_j by agent i be called C_i. For each considered question, the agent calculates the expected reward for answering this question. This expected reward, E_{i,q_j}, is simply the probability of answering the question correctly times the reward for doing so. This reward is dependent on the incentive mechanism being used by the system. For the control group which emulates the mechanism used in Yahoo! Answers, simply supplying an answer is worth 2 points, and 10 points are given for supplying the best answer. Additional points are given for the number of times that a user "likes" the answer. See table 2.1 for a full description of this mechanism. Calculating the probability of giving a best answer or the expected number of "likes" requires modeling every other agent in the system, and this is impractical for large systems. Therefore, when operating under the the Yahoo! Answers mechanism the simulation agents calculate E_{i,q_j} based on the expectation of getting the answer correct, and a correct answer is worth 1 additional point. Agents know their own expertise distributions, which is $\sim N(x_{\mu_i}, x_{\sigma_i})$, and the question difficulty d_j is a fixed number between 0 and 1. Therefore, the probability of getting the correct answer equals the probability of drawing a number z_{i,q_j} from their expertise distribution that is greater than the question difficulty, d_j. It is reasonable to assume that live users are capable of determining how well they are able to answer a given question. It is much more difficult and unlikely that users will know the probability of others giving correct answers to a question.

$$[\forall i \in I, \forall q_j \in C_i], E_{i,q_j} = 2 + P(z_{i,q_j} > d_j)$$

When operating under the experimental reciprocal mechanism developed in section 5.2, this expected reward E^*_{i,q_j} is now a function of the point total of the questioner, p_j, and a constant weight, ω.

$$[\forall i \in I, \forall q_j \in C_i], E^*_{i,q_j} = P(z_{i,q_j} > d_j)p_j\omega$$

Each agent then sorts all of the considered questions by expected reward and answers them starting with the highest expected reward. The agent stops answering questions when one of three criteria occurs:

1. All of the considered questions have been answered.

2. The expected reward for questions q_j becomes ≤ 0.

3. The agent has answered the maximum number of questions per cycle, a fixed constant M.

An answer is simply a number drawn from the expertise distribution of an agent. Rewards are calculated based on the difficulty of the question d_j, the quality of the answer, z_{i,q_j}, and depending on which incentive mechanism is used, the point total of the questioner, p_j. The answers for each question are then collected and evaluated. The best answer to a question is the answer with the highest value of those given for that question. The agent who supplied this best answer is given a bonus B of 10 points in the Yahoo! Answers mechanism, and a bonus B_j^* of five times the reward of an answer that is simply correct in the experimental mechanism. When the answer is not the best answer $B = 0$. These rewards R_{ij} for the control mechanism and R_{ij}^* for the experimental mechanism are expressed below.

$$R_{ij} = \begin{cases} 3 + B & : z_{i,q_j} \geq d_j \\ 2 & : z_{i,q_j} < d_j \end{cases} \tag{5.1}$$

$$R_{ij}^* = \begin{cases} p_j\omega + B_j^* & : z_{i,q_j} \geq d_j \\ 0 & : z_{i,q_j} < d_j \end{cases} \tag{5.2}$$

These points are then awarded to each user. In the Yahoo! Answers website, there is a problem of users copying content from the answers of others in an attempt to create the most comprehensive answer. This predatory behavior is called sniping. Such fraudulent behaviors are discussed further in section 5.4. One of the rules of the experimental mechanism is that users cannot see others' responses until the question has been closed and rewards distributed. This rule eliminates the threat of sniping. This simulation recreates Yahoo! Answers in a favorable light because sniping is not possible. Also, note that zero points are awarded in the experimental mechanism if the correct answer is not achieved. This is done to eliminate the incentive to create worthless, or spam answers.

In the control mechanism points are accumulated, and then they are spent when asking a question, see table 2.1. This discourages people from asking questions. Often those with the most expertise ask very few questions, if any. They do not want to risk their leader board standing. These are the most valuable people in the community, yet they are punished by the control incentive mechanism. In the experimental mechanism, asking questions is not discouraged, and there is no penalty for doing so. Under this mechanism, the relative difference in points accumulated has real value. Because their questions are "worth more", the leaders are given priority consideration when asking questions. In order to prevent this from becoming an exclusive club and discouraging new users from participating, new users are instantiated with a balance of 100 points, and point balances undergo a time decay in the experimental mechanism. In the simulation after each cycle the point balances are reduced by ϵ under the experimental mechanism.

A major obstacle that many online communities face is user attrition. In order to best model a real community operating in the steady-state, this simulation models the influx of new users and the defection of current users. The simulation built by Zhang, Ackerman, and Adamic in [89] models incoming new users until a certain graph density is achieved. This represents the bootstrapping problem of how an online community is formed, but it does not accurately capture the steady-state operation of a mature community. To simulate defection the agents are ranked according to point balance. The probability of agent i defecting $P(F_i)$ is then a function of their percentile rank t in the system. This ensures that the most successful agents are very unlikely to defect, while those who have difficulty accumulating points are much more likely to defect. This matches observed patterns on Yahoo! Answers and other online communities [39].

$$P(F_i) = \frac{1}{5\sqrt[4]{t}} - 0.2$$

Using this equation for calculating the probability of defection $P(F_i)$, the expected number of defectors can be calculated by solving the definite integral:

$$\int_0^1 \frac{1}{5\sqrt[4]{t}} - 0.2\,\mathrm{d}t = 0.066\bar{6}$$

This means that roughly 6.7% of the all the users will defect in any given cycle, and lower ranked users are much more likely to do so. This more accurately represents the behavior in a live system than simply eliminating the lowest performers. To balance this attrition, new users are introduced. The number of new agents, or users, added every cycle is determined by an integer that is $\sim N(0.07|I|, 0.01|I|)$. Recall that $|I|$ is the number of users in the system. These two equations representing defecting existing users and the creation of new users are balanced. This is designed to model the steady state operation of a QA system with very slow growth, and these parameters can be adjusted

Table 5.2: Simulation parameters

Name	**Symbol**	**Value**
Initial number of users	$\|I\|$	250
Expertise exponent	m	3
Expertise standard deviation multiplier	s	0.2
Question ask constant	α	10
Difficulty exponent	D	3
Question seen constant	β	2
Question exponent	K	0.5
Reciprocal reward weight	ω	0.1
Maximum number of answers	M	5
Point decay percentage	ϵ	5%

to model other scenarios. Table 5.2 summarizes the simulation parameters used for the experiments in section 5.3.2.

5.3.2 Simulation Results

The simulator was run in several different configurations for fifty complete cycles. One set of fifty cycles completes a single round. Results were then collected after twenty rounds have been completed. Reward points and users persist between cycles, but there is no concept of state that is preserved between cycles. Because we are most concerned with the behaviors of experts in this simulation, this section analyzes the performance of the top performing agents. Reward points are an artificial construct designed to encourage participation among the experts. Analyzing point accumulation alone is not meaningful, therefore the analysis presented here focuses on expertise and participation, which have a measurable impact on the usefulness of the com-

Table 5.3: Top 10% of point earners

	standard		reciprocal		reciprocal⋆	
Measurement	$\bar{X}$	$\hat{\sigma}$	$\bar{X}$	$\hat{\sigma}$	$\bar{X}$	$\hat{\sigma}$
Expertise mean, μ	0.745	0.151	0.838	0.118	0.886	0.0745
# Questions asked	0.941	0.957	0.569	0.869	0.462	0.776
# Answers received	66.75	42.25	62.46	52.29	83.41	83.23
# Questions answered	248.68	1.425	175.11	64.78	192.54	56.65

munity. Table 5.3 contains the data collected from the top 10% of the point earners after fifty cycles, averaged over twenty rounds.

Table 5.3 shows the performance of the top point earners. In other words, these measurements characterize the agents in the system that earned the highest rewards. This can be used to evaluate the incentive mechanism because this table describes the behaviors of most rewarded agents. The first configuration in Table 5.3, the "standard" column group, represents the performance characteristics of a generalized version of the Yahoo! Answers mechanism. This simulated mechanism is actually expected to perform better than the authentic version because the simulated version is immune to sniping and spam. The second column group contains the performance measurements of the reciprocal mechanism that is developed in section 5.2. The final column group, labeled "reciprocal⋆", duplicates the functionality of the "reciprocal" mechanism, but with the added component of an ideal recommender. Essentially this recommender mimics the functionality of an omniscient recommender because it allows agents to evaluate all questions and pick the most suitable ones to answer. Ordinarily an agent has a limited pool of questions under consideration, which models the human usage of a QA system. On a system of any appreciable size, no user has the ability or inclination to read every question. Section 4.1 discusses recommendation further.

Because each of the measurements in Table 5.3 represents data that are collected from twenty rounds of fifty cycles each, the measurements are expressed as a sample mean, $\bar{X}$, and a sample standard deviation $\hat{\sigma}$. The size of each sample is then equal to the number of rounds, in this case twenty.

The first row contains the average expertise mean, μ_i, for the agents in the top 10% of point earners. Recall that agents are instantiated with an expertise mean drawn from a power law, and it is bounded between zero and one, with lower values much more likely. Both the reciprocal and the reciprocal* mechanisms are more effective than the standard at rewarding the agents with the highest expertise. This difference is statistically significant according to an independent, two sample, two tailed t-test for statistical significance ($\alpha < 0.05$). Figure 5.5 illustrates this.

The next row shows the average number of questions asked by the top ranked agents. As expected, the top ranked agents do not ask many questions. This is because the probability of asking a question is inversely proportional the expertise level of the agent, and naturally the top earning agents are those with the most expertise. Due to the high sample standard deviation values the differences within this row are statistically significant only with a value of $\alpha < 0.2$

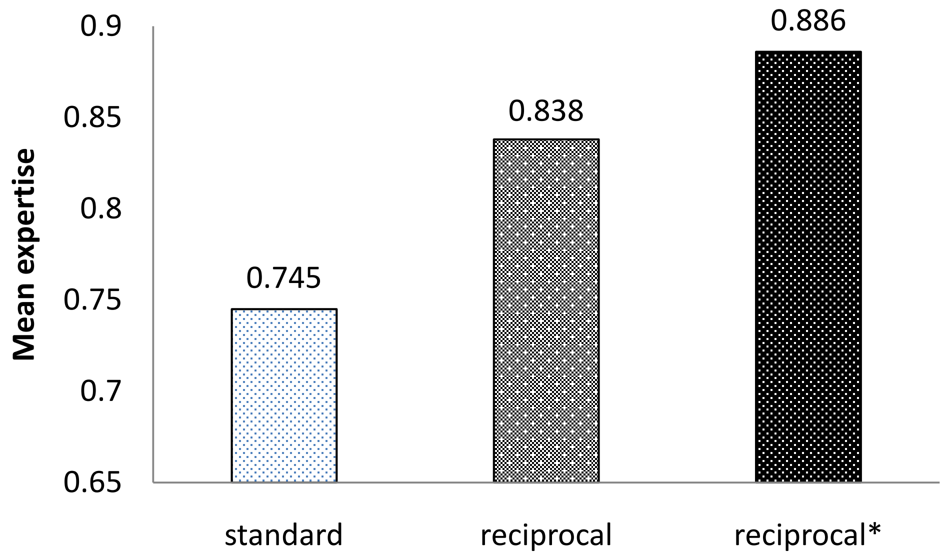

Figure 5.5: Mean expertise of the top 10% of point earners

The following row contains the average number of answers received per top ranked agent. Here again the reciprocal mechanisms perform at least as well as the standard mechanism. Consider that under the reciprocal mechanisms the agents asked roughly half as many questions as the agents adhering to the standard mechanism, yet they receive as many or more (in the case of reciprocal*) responses. Therefore, under the reciprocal mechanism the top performers receive twice as many responses for their questions. This is a key strength of the reciprocal mechanism. These extra responses comprise the systemic reciprocal reward.

The final row in Table 5.3 shows the number of questions answered by the top earning agents. The standard mechanism consistently yields a significantly larger number of questions answered than the other mechanisms, as

Table 5.4: Top 10% of experts, by μ_i

	standard		reciprocal		reciprocal*	
Measurement	$\bar{X}$	$\hat{\sigma}$	$\bar{X}$	$\hat{\sigma}$	$\bar{X}$	$\hat{\sigma}$
Expertise mean, μ	0.896	0.0585	0.941	0.0339	0.941	0.0342
# Questions asked	0.321	0.603	0.153	0.380	0.162	0.404
# Answers received	23.34	27.04	16.57	23.15	28.40	45.30
# Questions answered	146.55	98.56	133.87	86.07	142.31	85.99

indicated by the highest mean and small standard deviation. This is because the Yahoo! Answers mechanism rewards simply providing an answer, as indicated in table 2.1, regardless of correctness or if it comes from a reliable, expert source. Additionally, it is suspected that this is due to the bootstrapping dynamics of the experimental mechanisms. Under these reciprocal mechanisms, each agent starts on a level playing field, but it is possible to gain rewards more rapidly than in the standard model, causing fragmentation within the community. An agent with high expertise which fails several questions shortly after instantiation will be much more likely to defect from the system, as described in section 5.3.1. Ultimately, this means that some of the highest achievers are relative newcomers to the community, and they simply have not had the time to answer as many questions. Under the standard mechanism those with the most expertise slowly percolate to the top, and they tend to stay there for a long time and answer many questions.

Table 5.4 is very similar to table 5.3, but instead of measuring the agents from the top 10% of point earners, it contains measurements from the agents ranked in the top 10% according to expertise, μ_i. Previously, table 5.3

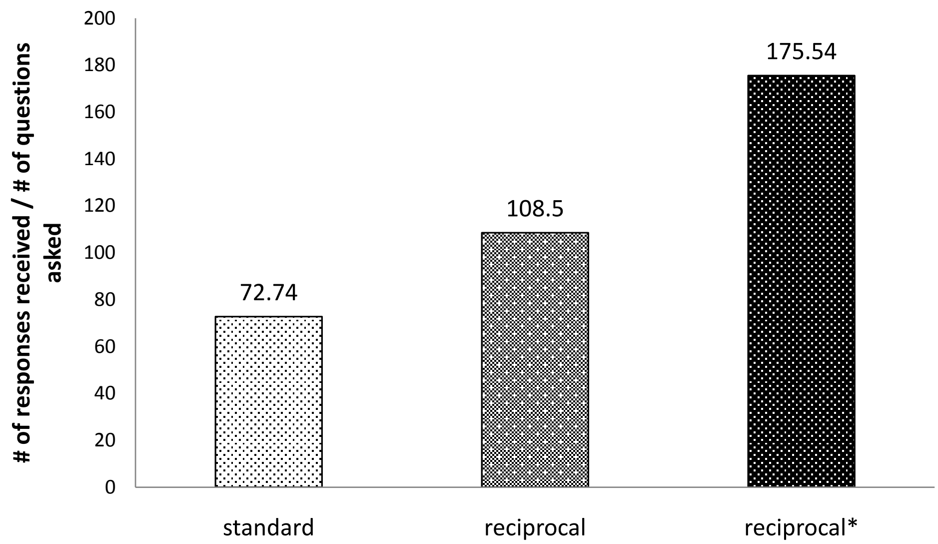

Figure 5.6: Answers received per questions asked among the top 10% by mean expertise

can be used to evaluate the incentive mechanism directly. Table 5.4 is perhaps even more interesting because it shows how the top experts in the system behave. The first thing to notice is the strong similarity between these two tables. The expertise means in the first row of Table 5.3 are close to the true maximum expertise means as shown in table 5.4. This assures us that all of the tested mechanisms are fairly effective at identifying the experts and that the reciprocal mechanisms outperform the standard mechanism. All mechanisms instantiate users in the same manner. The only remaining explanation for the discrepancy in expertise mean values (μ) between the different mechanisms is that the standard mechanism kept some of the best experts at an artificially low reward level and promoted lesser agents, thus increasing the chance of expert defection. Also notice in this table the difference in the number of answers received per question asked across the three mechanisms is even more apparent. This is illustrated in Figure 5.6. The final row in table 5.4 shows that the agents with the highest expertise do not answer significantly more questions under the standard mechanism. Therefore, the reciprocal mechanisms are just as good as the standard one for encouraging experts to answer questions. This section demonstrates that in a software simulation the reciprocal mechanism outperforms the standard Yahoo! Answers based mechanism according to several different metrics, including rewarding expertise and drawing a larger number of responses per question asked by experts, as shown in Figures 5.5 and 5.6.

5.4 Q&A Fraud

Because the reward of system influence has real value, users may be tempted to cheat the system to unfairly collect larger rewards. This hurts others in the system because the relative disparity in influence scores helps to drive others to participate. Thus, artificially inflated scores can unbalance the system.

Fraud can have many different forms. Often in aggregative systems a user may simply create spam for one of two reasons. First is to advertise an outside product or service in which the user has an interest. Thankfully we can rely on the collective wisdom of the other users to identify this behavior and nullify its effects. Another type of spam is the contribution of relevant but low quality content in an attempt to inflate one's participation score. From a system perspective this appears almost identical to a situation where a user with little expertise but much free time contributes much content at the best of his ability - a situation we wish to encourage. Fortunately this type of behavior is not rewarded as greatly as the case when a user demonstrates true expertise. This is due to the collaborative nature of the incentive mechanism. Endorsements from users with high expertise would allow greater influence achievement when contributing valuable content.

The incentive mechanism must be designed to combat various types of fraud. Some obvious fraud deterrents include harsh penalties for getting caught performing fraudulent behavior. Perhaps less obtrusive, this research combats fraud through the principle of incentive compatibility. A mechanism is said to be incentive compatible if every participant fares best (earns the most reward) by truthfully sharing private information, or in the context of QA, participating to the best of his ability. Though widely adopted, the mechanism behind Yahoo! Answers has a design flaw: it provides an incentive for answer *sniping*. Sniping occurs when a responder may not know the answer to a question and simply collects pertinent pieces of the answers from previous responders. Then this responder may have the most comprehensive answer, though he/she did not add any new content. This act of assembling the information does have some value, but rewarding this user instead of the original contributors does them a disservice. Jain, Chen, and Parkes state that this mechanism is not incentive compatible with a desired outcome because users have an incentive to snipe answers [40]. They suggest a rule where the asker distributes the rewards across multiple answer contributors. The mechanism developed in section 5.2 has a more elegant solution to this problem: responders cannot see the responses of others until the question answering period is completed, as decided by either by the original questioner or a system-wide policy based on activity and time. This simple change to the mechanics of a QA system eliminates the threat of answer sniping while maintaining author integrity.

Several types of fraud may be more difficult to prevent using incentive mechanism design. One consideration is that users may simply create positive feedback for themselves. Because the influence score is based on the feedback and influence of the rating user, a user who evaluates his content positively would create an infinite loop. This is thwarted in the proposed QA system by simply disallowing a user from rating his own content. This type of self-feedback can also be created in a more sophisticated manner by collusive voting. Either multiple users may set up secret agreements to provide positive feedback, or a single user may have multiple accounts which evaluate each other. Trust can be applied at the application level to detect this type of fraud. Users who provide ratings that are not corroborated by others may be identified as fraudulent. If it can be observed that two or more users regularly give each other positive feedback in the absence of positive feedback from others, it is possible that this type of fraud has occurred. Such sophisticated techniques for detecting and mitigating fraud are outside the scope of this dissertation.

Perhaps an even more difficult type of fraud to detect is shared accounts. From a system perspective, multiple users on the same account would appear as a single user with very high participation and a broad body of expertise. This could enable rapid growth of influence, and each person could reap the benefit of asking questions under this username. It is likely that many questions of very diverse topics and length would originate from this single user account. This could possibly be detected by analyzing the user's question to answer ratio or the question topic diversity. On the network level this type of fraud may become more apparent by analyzing the IP address of the content origin. A shared account would likely have simultaneous people logged into the account from many different IP addresses.

Fraud has the capability to cripple an online community. It is imperative to combat this fraud with all necessary means while maintaining system functionality. Application level solutions, like trust, can be combined with network level solutions, like traffic and IP monitoring, to detect fraudulent behavior. The most important step toward fighting fraud is to remove any incentive to perform fraudulent behavior through careful mechanism design.

5.5 Chapter Summary

This chapter defines the task of mechanism design, investigates a new class of incentives, develops an incentive compatible mechanism for question and answer systems, and tests the mechanism empirically through a software simulation. This incentive mechanism is designed to plug into the "incentive calculator" module of the QA system architecture presented in chapter 3. The results in section 5.3.2 above show that the new, experimental mechanism modestly, yet significantly improves upon the industry standard model. This simulation compares the mechanisms solely as an allocation of points based on behaviors in the system. However, these benefits are magnified when the mechanism operates on a QA system in tandem with a content recommender. Combined with a new incentive that has been shown to be of value to live users, these advances promise to create communities capable of promoting and rewarding the most valuable contributors.

Recall $RQ2$ from section 1.2:

> *RQ 2: How effective are systemic reciprocal rewards for encouraging expert participation in an online community?*

The survey in section 5.1.1 shows that people are receptive to new classes of rewards beyond badges and trophies. Specifically they prefer rewards that have value within the community, or systemic rewards.

Recall $RQ3$ from section 1.2:

> *RQ 3: In the context of a question and answer system, how can an incentive mechanism be designed to encourage expert participation?*

Section 5.2 details the design and development of a new, reciprocal incentive mechanism, and section 5.3.2 characterizes its performance in a software simulation.

Chapter 6

Impact

The purpose of this research is to discover how to encourage expert participation in online communities. These communities are growing rapidly and we have come to rely on them as a source of valuable information and entertainment. They can take many forms, including a question and answer system, a news aggregation service, a discussion forum, or a social network, just to name a few. Most of the research presented here pertains to QA systems as an example, but it is adaptable to other forms on online communities as well. Experts across these various communities are those who add the most value to the community, therefore their participation is highly desired. Encouraging expert participation can be decomposed into three constituent problems.

1. Identifying the experts

2. Choosing a reward

3. Distributing the reward

Each of these subproblems has been discussed in detail, and novel techniques for accomplishing these tasks have been developed. These three problems correspond directly to the original three research questions presented in section 1.2. Recall research question 1:

> *RQ 1: How can the expertise of a user in an online community be measured?*

The expertise models introduced in chapter 4 allow one to quantitatively measure the expertise of a user in an online community. The concept of topics as distributions over words as presented in section 4.1 allows the characterization of the breadth of a user's expertise in a QA system. In other words, this technology allows us to identify a user's areas of experience and interest. The new generative model based approach of measuring expertise has been shown to consistently outperform standard information retrieval approaches and sometimes outperform clustering algorithms when identifying topics of interest for a user in a QA system. The following section, 4.2, describes an algorithm for measuring the depth of expertise, or reputation, of users in any content oriented social network (COSN) in general. This algorithm, rooted in the trust in multi-agent systems community, has been shown to outperform common measures of authority, such as PageRank [63], in networks that are prone to negative social effects. These two sections in chapter 4 introduce new technologies to measure both breadth and depth of expertise in online communities, and they are shown to perform as well or better than the established methods.

The second constituent problem listed above is formulated in research question 2:

> *RQ 2: How effective are systemic reciprocal rewards for encouraging expert participation in an online community?*

A systemic reward is one that has value within the framework of the system, or community. Such rewards are given in exchange for contributing valuable expertise to the community. They are designed to give extra functionality, enjoyment, or ease of use to the awardee. By their very nature they must be distributed in a competitive manner, otherwise they would simply be implemented in the community for all users to take advantage of. For example, priority access to information or users has no value if all users are given priority. This reward is said to be *reciprocal* because the reward gained by a content creator is a function of the social influence of the user who requested the content. Systemic reciprocal rewards are a new class of rewards for online communities. The standard reward model in industry is an achievement-based model where users gain virtual prizes and leader board standings for their participation. Beyond bragging rights these rewards have no value. Section 5.1 presents a survey measuring the reward preference of 380 volunteers. The results unequivocally show a strong preference for systemic reciprocal rewards over the industry standard rewards, regardless of participant demographic information. This research is the first to show that access to peer generated content can directly motivate people to apply their own expertise, thereby generating more content.

The third constituent problem listed above is formulated in research question 3:

RQ 3: In the context of a question and answer system, how can an incentive mechanism be designed to encourage expert participation?

Section 5.1 presents a suitable reward for expert participation, and section 5.2 describes the design of a full mechanism based on such a reward and analyzes its performance in a software simulation of a QA system. The experimental, or reciprocal, mechanism has been shown to more effectively reward and retain the experts in a community than the industry standard point-based mechanism. In addition to this benefit, the experts themselves received more answers when asking their own questions under the reciprocal incentive mechanism. Section 5.2 also shows that ideal recommendation only magnifies these effects. Moreover, the reciprocal mechanism eliminates the incentive to create spam and snipe answers, and it does not discourage asking questions. Because of these properties the incentive mechanism developed in this research is superior to the industry standard for encouraging expert participation, and this has been empirically confirmed in a software simulation based on observed human user behavior.

Chapter 3 describes an architecture for QA that is used as a framework for developing three of the primary contributions of this research: a multi-dimensional topic-specific model of expertise, the concept of systemic reciprocal rewards, and a full implementation of an influence-based incentive mechanism. Recall the research hypothesis from chapter 1:

> *In an online community, a non-monetary quantitative incentive mechanism applied to a recommendation-based architecture can increase expert participation and satisfaction - both as a content creator and a consumer - and assure confidence in the value of the provided answers.*

The three research questions in section 1.2 are designed to test this hypothesis. The hypothesis states that an expert user, both as a content creator and a consumer, will have increased participation and satisfaction from an online community that is built upon the technology developed in this dissertation. As a content creator in a QA system, recommendation as defined in section 4.1 increases expert satisfaction because recommendation lowers the barrier of entry to providing expertise in the form of answering questions. This recommendation process ensures that content creators can be occupied by answering questions from within their areas of interest, thereby creating an intrinsic reward for participation and increasing satisfaction. This has been explored in the work surrounding research question 1. Work supporting research questions 2 and 3 develop extrinsic motivators based on the concept of systemic reciprocal rewards. These rewards are developed in chapter 5 and shown to be vastly preferred over the standard model in section 5.1. These incentives are designed to increase the participation of experts as content creators.

In a QA system a consumer is one who asks questions or uses the pool of answered questions as a reference. As a consumer, recommendation as described in section 4.1 increases expert satisfaction by assuring the user that questions have been seen by the most appropriate experts. A possible extension to this recommendation is to provide the consumer with a quantitative measure of the respondents' expertise and how well it matches a given question. As a consumer, participation is increased by the proliferation of experts and the availability of valuable content within the community.

The expertise models developed in chapter 4 demonstrate the capability of identifying experts in various types of communities based on the content that is contributed by the user and also the link structure that emerges from the social features of various types of online communities. These expertise models facilitate recommendation, which acts as an intrinsic motivator toward participation because it helps to make the process of answering questions easy and enjoyable. Addressing the final research question concerning mechanism design, section 5.2 contains the design and development of a novel mechanism that is tested in a software simulation presented in section 5.3.

A promising area for future work involves examining the behavior of live users operating under the novel incentive mechanism. This work shows that live survey respondents have expressed preference for such a mechanism, but this is not necessarily an indicator of how they would behave in a live system. A test in a real-world environment would make a more convincing argument for a new class of incentives. Additionally, research on a live system would allow further development toward an optimal mechanism. The parameters of the simulation in section 5.3, while based on observations and analysis of existing

QA forums, have been chosen to best approximate generalized behavior in QA systems. Mechanism optimization is highly domain dependent, and different QA systems have very different dynamic behavior.

A natural continuation of this work involves adapting this class of incentive mechanisms to other types of online communities. The foundation of adapting expertise models to news aggregation sites and content oriented social networks in general is demonstrated in section 4.2. An adaptation to *networking* oriented social networks such as Facebook and LinkedIn would be particularly interesting. Currently these sites rely primarily on intrinsic rewards; linking to someone is its own reward. A layer of incentives on top of such communities could spur future growth.

An online Question and Answer system is a prime example of a peer production system [49]. In a QA system, a self-organized community emerges, and the producers of content are also the consumers. One of the greatest strengths of such systems, as well as a potential criticism, is the democratization of knowledge (DOK).

DOK describes the way knowledge is constructed and disseminated. A body of public knowledge can be manipulated by the people, not necessarily experts. Often this is seen as a good thing, such as the sharing of education materials by universities [25]. A benefit of this is that knowledge is no longer kept secret. Via the internet, people with specialized knowledge are able to garner a larger audience than possible before. It is revolutionizing the way course materials are disseminated at top universities [25]. Moreover, knowledge can be contributed in a distributed fashion, giving everyone the ability to share and modify the knowledge originating from others (e.g. Wikipedia).

The QA system discussed in this dissertation, along with other types of online communities, supports these positive traits of DOK by sharing the results of all historic and current questions and their respective answers with all users, regardless of participation. Everybody has access to the information, and everyone is capable of adding to the information by providing an answer.

Some serious drawbacks to DOK exist. According to Hofstadter, DOK encourages anti-intellectualism and utilitarianism [35]. Much like *design by committee*, DOK can result in banal knowledge findings, since people must have agreement or it will likely be changed. Just like *design by committee* automobiles end up looking amorphous, DOK can result in knowledge that is least objectionable or *safe*. This makes it difficult to express controversial concepts, and opinions must be surrounded by the appropriate caveats. Moreover, this encourages noise, or the dissemination of knowledge that everyone already knows.

The online communities supported by this research combat these drawbacks by preserving author identity. Experts are recognized, and their contributions are rewarded. An author is accountable for his work, and there are consequences of providing useful content, through influence advancement, and also for providing negligible content, through wasted time and effort. Because anyone can create an answer and read existing answers, but each answer is the work of one identified contributor, a QA system is able to leverage the positive aspects of democratization of knowledge while avoiding the negative aspects such as censorship and banality.

This dissertation proposes an engineering solution to a fundamentally human problem. The proposed expertise modeling process, recommendation architecture, and incentive mechanism are designed to lower the barrier to entry and encourage expert participation in online communities. Increased expert participation ensures added value, and in the context of QA systems, accurate solutions and satisfied questioners. The impact of this work extends beyond QA and applies to peer production systems in general. The research presented here is the first to show how content generated by peers, with no intermediate monetary value, can directly motivate people to apply their expertise and effort toward a socially beneficial system.

Index

Bibliography

[1] Aardvark. Aardvark social search, November 2009. `http://vark.com`.

[2] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 665–674, New York, NY, USA, 2008. ACM.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, June 2005.

[4] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 183–194, New York, NY, USA, 2008. ACM.

[5] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW*

'08: Proceeding of the 17th international conference on World Wide Web, pages 199–208, New York, NY, USA, 2008. ACM.

[6] Olle Bälter and Katarina Augustsson Bälter. Demands on web survey tools for epidemiological research. *European Journal of Epidemiology*, 20(2):137–139, February 2005.

[7] K. Barber and J. Kim. Belief revision process based on trust: Agents evaluating reputation of information sources. *Trust in Cyber-societies*, pages 73–82, 2001.

[8] Y. Benkler. Coase's Penguin, or, Linux and the Nature of the Firm. *Yale Law Journal*, 112(3):367–445, 2002.

[9] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007. Citeseer, 2007.

[10] Klaus Berberich, Srikanta Bedathur, Gerhard Weikum, and Michalis Vazirgiannis. Comparing apples and oranges: normalized pagerank for evolving graphs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1145–1146, New York, NY, USA, 2007. ACM.

[11] Thomas J. Bergmann and William J. Hannaford. Amount, timing, and value of financial incentives in mail surveys: Does it make a difference? *Marketing Research*, 2(3):30 – 36, 1990.

[12] BlurtIt. Blurtit.com, November 2009.
`http://www.blurtit.com`.

[13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.

[14] Roger Browne. Google Answers: Google's Question and Answer service from 2002-2006, September 2008.
`http://knol.google.com/k/roger-browne/google-answers/1xdiqvw5h03s1/2#`.

[15] Suratna Budalakoti, David DeAngelis, and K. Suzanne Barber. Expertise modeling and recommendation in online question and answer forums. In *SiN '09: Proceedings of the IEEE Conference on Social Computing, Symposium on Social Intelligence and Networking*, Vancouver, BC, Canada, 2009.

[16] Suratna Budalakoti, David DeAngelis, and K Suzanne Barber. Unbiased Trust Estimation in Content-Oriented Social Networks. In *Proceedings of The Workshop on Trust in Agent Societies at The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2011)*, Taipei, Taiwan, 2011.

[17] Y. Chen, T.H. Ho, and Y.M. Kim. Knowledge Market Design: A Field Experiment at Google Answers.

[18] Allan H. Church. Estimating the effect of incentives on mail survey response rates: A meta-analysis. *The Public Opinion Quarterly*, 57(1):62–79, 1993.

[19] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 160–168, New York, NY, USA, 2008. ACM.

[20] Adam D'Angelo and Charlie Cheever. Quora.com, November 2009. `http://www.quora.com`.

[21] Matthew A. Wormley Daniel K. Gardner, Clinton A Staley. Question and Answer System Using Computer Networks. Experts Exchange, Inc., May 2000. `United States Patent 6,064,978`.

[22] D. DeAngelis, K. Fullam, and K.S. Barber. Effects of Communication Disruption in Mobile Agent Trust Assessments for Distributed Security. In *Proceedings of The Workshop on Trust in Agent Societies at The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005), Utrecht, July*, pages 25–29. Citeseer, 2005.

[23] AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[24] P. Edwards, I. Roberts, M. Clarke, C. DiGuiseppi, S. Pratap, R. Wentz, and I. Kwan. Increasing response rates to postal questionnaires: systematic review. *British Medical Journal*, 324(7347):1183, 2002.

[25] eSchool News. Web Fuels democratization of knowledge, March 2007. `http://www.eschoolnews.com/news/top-news/index.cfm?i=45786`.

[26] Marian Gallagher. Questions and Answers. *Law Library Journal*, 46:93–97, 1953.

[27] Rumi Ghosh and Kristina Lerman. Predicting influential users in online social networks. In *Proceedings of KDD workshop on Social Network Analysis (SNA-KDD)*, July 2010.

[28] J.R. Gibb. *Trust: A new view of personal and organizational development*. Guild of Tutors Pr, 1978.

[29] H. Godil. *Finding Experts by Modeling Domain Expertise*. University of Texas at Austin, 2006.

[30] N. Good, J.B. Schafer, J.A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 439–446. JOHN WILEY & SONS LTD, 1999.

[31] Nishit Gujral, David DeAngelis, Karen Fullam, and K. Suzanne Barber. Modeling Multi-Dimensional Trust. In *the Proceedings of the Workshop on Trust in Agent Societies*, pages 8–12, 2006.

[32] Lei Guo, Enhua Tan, Songqing Chen, Xiaodong Zhang, and Yihong (Eric) Zhao. Analyzing patterns of user content generation in online social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 369, New York, New York, USA, 2009. ACM Press.

[33] J.A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc. New York, NY, USA, 1975.

[34] Taher Haveliwala, Sepandar Kamvar, and Glen Jeh. An analytical comparison of approaches to personalizing pagerank. Technical Report 2003-35, Stanford InfoLab, June 2003.

[35] R. Hofstadter. *Anti-intellectualism in American life*. Knopf, 1963.

[36] Leonid Hurwicz and Stanley Reiter. *Designing Economic Mechanisms*. Cambridge University Press, 2008.

[37] Experts-Exchange Inc. Experts-Exchange About Us, August 2009. `http://www.experts-exchange.com/aboutUs.jsp`.

[38] Mahalo Inc. Mahalo Answers, November 2009. `http://www.mahalo.com/answers/`.

[39] Yahoo! Inc. Yahoo! Answers, November 2009. `http://answers.yahoo.com`.

[40] Shaili Jain, Yiling Chen, and David C. Parkes. Designing incentives for online question and answer forums. In *EC '09: Proceedings of the tenth ACM conference on Electronic commerce*, pages 129–138, New York, NY, USA, 2009. ACM.

[41] Catholijn M. Jonker and Jan Treur. Formal analysis of models for the dynamics of trust based on experiences. In *Multi-Agent System Engineering*, volume 1647 of *Lecture Notes in Computer Science*, pages 221–231. Springer Berlin / Heidelberg, 1999.

[42] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 919–922, New York, NY, USA, 2007. ACM.

[43] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[44] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[45] John George Kemeny and James Laurie Snell. *Finite Markov chains.* University series in undergraduate mathematics. VanNostrand, New York, repr edition, 1969.

[46] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[47] P. Kollock. The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace. *Communities in cyberspace*, page 220, 1999.

[48] Peter Kollock. "an eye for an eye leaves everyone blind": Cooperation and accounting systems. *American Sociological Review*, 58(6):768–786, 1993.

[49] C. Leadbeater. *We-think: The power of mass creativity.* Profile, 2008.

[50] J.G. Lee, P. Antoniadis, and K. Salamatian. Faving Reciprocity in Content Sharing Communities: A Comparative Analysis of Flickr and Twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 136–143. IEEE, 2010.

[51] Jacob Leibenluft. A Librarian's Worst Nightmare: Yahoo! Answers, where 120 million users can be wrong. *Slate Magazine*, 2007.

[52] Kristina Lerman and Rumi Ghosh. Information contagion: an empirical study of spread of news on digg and twitter social networks. In *Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM)*, May 2010.

[53] Kristina Lerman and Laurie Jones. Social browsing on flickr. In *Proceedings of 1st International Conference on Weblogs and Social Media (ICWSM-07)*, 2007.

[54] J. Lussier, Troy Raeder, and N. Chawla. User Generated Content Consumption and Social Networking in Knowledge-Sharing OSNs. *Advances in Social Computing*, pages 228–237, 2010.

[55] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design Lessons from the Fastest Q&A Site in the West. In *CHI*, 2011.

[56] A.H. Maslow. A theory of human motivation. *Psychological review*, 50(4):370–396, 1943.

[57] J.A. Mazanec and H. Strasser. *A nonparametric approach to perceptions-based market segmentation: Foundations*, volume 1. Springer Verlag, 2000.

[58] J. Scott Mizes, E. Louis Fleece, and Cindy Roos. Incentives for increasing return rates: Magnitude levels, response bias, amd format. *The Public Opinion Quarterly*, 48(4):794–800, 1984.

[59] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. The structure and performance of an open-domain question answering system. In *ANNUAL MEETING-ASSOCIATION FOR COM-*

PUTATIONAL LINGUISTICS, volume 38, pages 563–570. UNKNOWN, 2000.

[60] Lik Mui, Ari Halberstadt, and Mojdeh Mohtashemi. Evaluating reputation in multi-agents systems. In *Trust, Reputation, and Security: Theories and Practice*, volume 2631 of *Lecture Notes in Computer Science*, pages 183–194. Springer Berlin / Heidelberg, 2003.

[61] R.B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58, 1981.

[62] Anton J. Nederhof. The effects of material incentives in mail surveys: Two studies. *The Public Opinion Quarterly*, 47(1):103–111, 1983.

[63] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1998.

[64] D.C. Parkes. *Iterative combinatorial auctions: Achieving economic and computational efficiency*. PhD thesis, Citeseer, 2001.

[65] E. Pavlov, J.S. Rosenschein, and Z. Topol. Supporting privacy in decentralized additive reputation systems. *Trust Management*, pages 108–119, 2004.

[66] Johan A. Pouwelse, Pawel Garbacki, Dick Epema, and Henk Sips. Pirates and samaritans: A decade of measurements on peer production and their implications for net neutrality and copyright. *Telecommunications*

Policy, 32(11):701 – 712, 2008. Future Telecommunications Regulation: A Dutch Perspective - Future Telecommunications Regulation.

[67] Quantcast. Experts-Exchange.com - Quantcast Audience Profile, August 2009. `http://www.quantcast.com/experts-exchange.com`.

[68] S. Reiss. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of General Psychology*, 8(3):179–193, 2004.

[69] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[70] Kevin Rose. Digg v4: release, iterate, repeat., July 2010.

[71] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[72] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 475–482. ACM, 2002.

[73] T. Sandholm. Automated mechanism design: A new application area for search algorithms. *Lecture notes in computer science*, pages 19–36, 2003.

[74] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 158–167, New York, NY, USA, 2000. ACM.

[75] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.

[76] G. Slutzki and O. Volij. Ranking participants in generalized tournaments. *International Journal of Game Theory*, 33(2):255–270, 2005.

[77] Yahoo! Answers Team. Yahoo! Answers blog: Did you know?, November 2009.
`http://yanswersblog.com/index.php/archives/2009/10/05`
`/did-you-know/`.

[78] N. Thurman. Forums for citizen journalists? Adoption of user generated content initiatives by online news media. *New Media & Society*, 10(1):139, 2008.

[79] Uclue. Uclue.com, November 2009.
`http://uclue.com`.

[80] Marshall Van Alstyne and Erik Brynjolfsson. Global Village or Cyber-Balkans? Modeling and Measuring the Integration of Electronic Communities. *Management Science*, 51(6):851–868, June 2005.

[81] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[82] Mahmoud A. Wahba and Lawrence G. Bridwell. Maslow reconsidered: A review of research on the need hierarchy theory. *Organizational Behavior and Human Performance*, 15(2):212 – 240, 1976.

[83] Howard T. Welser, Eric Gleave, Vladimir Barash, Marc Smith, and Jessica Meckes. Whither the experts? social affordances and the cultivation of experts in community q&a systems. *Computational Science and Engineering, IEEE International Conference on*, 4:450–455, 2009.

[84] H.T. Welser, E. Gleave, D. Fisher, and M. Smith. Visualizing the signatures of social roles in online discussion groups. *Journal of Social Structure*, 8(2), 2007.

[85] Jianshu Weng, E.P. Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[86] Robert W. White. Motivation reconsidered: The concept of competence. *Psychological Review*, 66(5):297 – 333, 1959.

[87] Jiang Yang, Lada A. Adamic, and Mark S. Ackerman. Crowdsourcing and knowledge sharing: strategic user behavior on taskcn. In *EC '08:*

Proceedings of the 9th ACM conference on Electronic commerce, pages 246–255, New York, NY, USA, 2008. ACM.

[88] P. Yolum and M.P. Singh. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35(3):396–407, 2005.

[89] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 221–230, New York, NY, USA, 2007. ACM.

[90] Jun Zhang, Mark S. Ackerman, Lada Adamic, and Kevin Kyung Nam. Qume: a mechanism to support expertise finding in online help-seeking communities. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 111–114, New York, NY, USA, 2007. ACM.

Vita

David DeAngelis was born in Lorain, Ohio, on 29 November 1981, the son of Frank DeAngelis and Janice DeAngelis. He completed his secondary education at Amherst Steele High School in Amherst, Ohio, in 2000. In 2004, David earned the degree of Bachelor of Science with College and University Honors from Carnegie Mellon University in Pittsburgh, Pennsylvania. He entered The Graduate School at the University of Texas at Austin in August 2004, supported by an Electrical and Computer Engineering Teaching Assistantship and the College of Engineering Doctoral Fellowship. He earned his Master of Science in Engineering degree in August of 2006 for his work titled *Security Applications of Trust in Multi-Agent Systems*.

Permanent address: 6344 Balsam Drive
Amherst, Ohio 44001
deangelis@utexas.edu

This dissertation was typeset with LaTeX† by the author.

†LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.